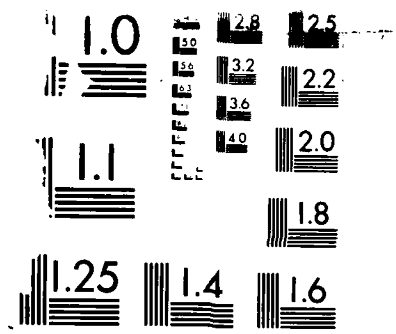
AD-A175 489 CLOUD MICROPHYSICS ANALYSIS AND MODELING(U) OPHIR CORP LAKEWOOD CO L D NELSON ET AL. JUL 86 AFGL-TR-86-0179 F19628-83-C-0130 1/2

UNCLASSIFIED F/G 4/2 NL



MICROCOPY RESOLUTION TEST CHART

(12)

AFGL-TR-86-0179

CLOUD MICROPHYSICS ANALYSIS AND MODELING

Loren D. Nelson
Keeley R. Hanson
Joshua M. Wurman
Raina J. Eckhardt

OPHIR Corporation
7333 West Jefferson Avenue, Suite 210
Lakewood, CO 80235

July 1986

Final Report
29 June 1983 - 30 June 1986

Approved for Public Release, Distribution Unlimited

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MA 01731-5000

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
n/a

A179 489

2a. SECURITY CLASSIFICATION AUTHORITY
n/a

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
n/a

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
n/a

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFGL-TR-86-0179

6a. NAME OF PERFORMING ORGANIZATION
OPHIR Corporation

6b. OFFICE SYMBOL *(If applicable)*
7U702

7a. NAME OF MONITORING ORGANIZATION
Air Force Geophysics Laboratory

6c. ADDRESS *(City, State and ZIP Code)*
7333 West Jefferson Avenue, Suite 210
Lakewood, CO 80235

7b. ADDRESS *(City, State and ZIP Code)*
Hanscom AFB, MA 01731-5000
Monitor/Barbara A. Main/LYC

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Air Force Geophysics Laboratory

8b. OFFICE SYMBOL *(If applicable)*
AFGL/LYC

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F19628-83-C-0130

8c. ADDRESS *(City, State and ZIP Code)*
Hanscom AFB, MA 01731-5000

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
62101F	6670	12	CA

11. TITLE *(Include Security Classification)*
CLOUD MICROPHYSICS ANALYSIS AND MODELING

12. PERSONAL AUTHOR(S)
Loren D. Nelson, Keeley R. Hanson, Joshua M. Wurman, Raina J. Eckhardt

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 6/29/83 TO 6/30/86

14. DATE OF REPORT *(Yr., Mo., Day)*
1986 July

15. PAGE COUNT
120

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
cloud physics, atmospheric modeling, atmospheric analysis, mesoscale modeling

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This final report documents the results of a three year contract effort between OPHIR Corporation and the Air Force Geophysics Laboratory to develop mesoscale cloud models and analyze data in support of such models.

Results of the BIGHILL and CCOPE experiment model runs and the transfer and implementation of the CSU Cloud and Mesoscale Model at AFGL and the AFWL computer systems are documented.

Operational instructions for the model and supporting file transfer and graphical utilities are included.

The NCAR graphics software system was modified and installed on the AFWL Cray computer. Test plots, metacode translator software, and operational instructions are provided. File transfer utilities such as kermit, and archival procedures to transfer data from the AFWL Cray, the AFGL VAX, and Zenith Z-100 personal computers are attached.

Other computer models and codes described include the CUMOD microphysical model of cumulus clouds, the SNOW-II snowstorm analysis, STRATEX data analysis, UND data analysis of aircraft flight tapes, and programs for the analysis of PMS-2D Particle Image Data.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Barbara A. Main

22b. TELEPHONE NUMBER *(Include Area Code)*
(617) 377-2947

22c. OFFICE SYMBOL
LYC

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

Geography plotting routines for use with the stratocumulus model are described and a complete Plot Routine Library for scientific data analysis on the AFGL VAX is documented in the form of a users manual.

~~Finally,~~ a series of data analysis programs for use in the 1986 AFGL Melting Layer Attenuation Study Analysis is described and documented.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

OPHIR provided research support to the Cloud Physics Branch in the Atmospheric Sciences Division of the Air Force Geophysics Laboratory. This research support included contributions to the development and application of models of the microphysics and dynamics of clouds.

The research also included development of a graphics analysis package, analysis of the model results, and of observational data from various sources such as instrumented research aircraft, radars, and surface instrumentation.

Existing computer programs were modified. Extensive development of new software was also provided.

2. RESULTS OF THE CLOUD AND MESOSCALE MODEL EXPERIMENTS

The model used by the Cloud Physics Branch of the Atmospheric Sciences Division of the Air Force Geophysics Laboratory (AFGL) for their mesoscale studies was developed at Colorado State University by Tripoli and Cotton (Tripoli and Cotton, 1982 and Cotton et. al., 1982). The model is now called the Regional Atmospheric Modeling System (RAMS). The experimental results of the studies using the CSU RAMS model have led to some important conclusions in the field of atmospheric science. These experiments are briefly described below.

2.1 The BIGHILL Experiments

The initiation of cumulus clouds in the mountains remains one of the research interests of the Cloud Physics Branch. Previous studies had indicated several mechanisms for the development of cumulus clouds in complex terrain. The BIGHILL model experiments in which OPHIR was involved were designed to investigate one of these mechanisms, the leeside convergence zone. The existence of the leeside convergence zone had been simulated in model experiments by Dr. Robert Banta of the Cloud Physics Branch using a small hill and dry conditions. To test the role of the leeside convergence zone in the formation of cumulus clouds, the BIGHILL experiments were created. The BIGHILL study used a larger hill than the bump2d studies of Dr. Banta and included microphysical processes so the development of a cloud could be simulated.

The BIGHILL experiments were two dimensional and used version 2 of the CSU RAMS model. The mountain ridge was about 1 km high and was simple yet roughly shaped to represent the ridge to the west of South Park, CO. The initial sounding of temperature and moisture resembled that of 3 August 1977, a day without precipitation, in South Park. Ambient westerly winds of 2.5 m s^{-1} were assumed.

The first three model simulations were intended to illustrate the effects of varying the surface heat flux and the effects of large scale convergence on the formation and development of cumulus clouds. These first three simulations were called BIGHILL, BIGHTD, and BIGCNV. BIGHILL and BIGCNV used a surface heat flux of 17°K cm s^{-1}, and BIGHTD had a surface heat flux of 25°K cm s^{-1}. BIGCNV included domain scale convergence, represented by vertical velocities which varied with height and were of 10 cm s^{-1} or less. Warm cloud microphysical processes were included.

The results of these studies were presented at the Seventh Conference on Numerical Weather Prediction (Banta and Hanson, 1985). In each of the three simulations, easterly upslope winds developed and converged with westerly ridgetop winds just to the leeside of the hilltop. A convergence zone resulted and a cloud did develop in each of the three model runs. In the BIGHILL case, the cloud was only a small strato-cumulus type cloud which developed near the end of the 5 hour simulation. Moisture from near the surface had been advected upward by the updraft of the convergence zone, but the ambient winds had dispersed the moisture before an organized cumulus cloud had evolved.

In the BIGHTD case of the greater surface heat flux, a stronger convergence zone and updraft forced more surface moisture upward. Condensation took place after two hours and 23 minutes and the cloud continued to grow to an active cumulus cloud as it moved down the slope.

The updraft was also stronger in the case with domain scale convergence. The first cloud appeared in the BIGCNV simulation after two hours and 15 minutes of model time. By five hours of model time, the cloud

had grown very large and had propagated down the slope.

The results from the first three studies show that in the leeside convergence mechanism for cumulus cloud initiation, the balance between the rate of moistening of the upper portions of the deep convective boundary layer and the rate of the ventilation by the ambient winds controls how clouds form. The surface heat flux, in turn, controls how rapidly the convective boundary layer moistens, by driving the upslope flow which feeds the updrafts. A heat flux of 17°K cm s^{-1} was sufficient to produce only a thin strato-cumulus cloud, while 25°K cm s^{-1} did produce a convective cloud.

Another process which affects the intensity of the updrafts produced by mountain circulations is large-scale convergence. The case which included domain scale convergence produced deeper convection, sooner than either the weak or the strong synoptic surface heating cases. Therefore, larger scale synoptic conditions will play a very important role in the development of locally forced cumulus convection in complex terrain.

A fourth simulation BIGINT JOB A was initialized with no cold pool in the valley and without surface heat and moisture fluxes. The wind profile was initially defined to increase with height similar to the observed early morning winds in South Park. The purpose of this simulation was to learn how much of the convergence and updraft observed in the simulations might have been the result of dynamic forcing of a wave if the winds had been introduced into the domain too quickly. This case was run without forcing from surface heat and moisture fluxes until a steady state seemed to be reached (by 10800 seconds the equivalent of three hours). The results indicated that the higher wind speeds of the upper levels came down to the surface near the foot of the hill.

Another simulation BIGNOF JOB A was run with a different wind profile to reduce the initial vertical wind shear. It used the original, shorter domain and, to save on computer costs, the microphysical options were turned off. There was an area of convergence and updraft produced in this simulation without surface heating. The results also hinted that some of the higher wind speeds from upper levels had propagated down to the lower levels but the effect was less pronounced than in the previous case.

Each of the jobs BIGHILL JOB A, BIGHTD JOB A, and BIGCNV JOB A were run again with the ambient winds introduced into the domain more slowly. The winds were introduced over a period of 30 min with an additional 30 min to allow the winds to settle before the surface heating and large scale convergence was begun. Fourth order advection and an initial cloud droplet concentration of 850 cm^{-3} were included in these cases. Preliminary analysis of the results indicated that the differences between the weak and strong surface heating cases was more dramatic. The case with small surface heating led to weak convergence and an updraft which was not capable of producing a cloud. The large surface heating case not only had strong enough convergence and updraft to produce a cloud, but the cloud and the updraft region moved down the slope and developed rain like some observed mountain cumulus.

The case with the large scale convergence and the slower wind addition into the domain also utilized a taller domain and included ice and graupel microphysical processes. The results indicated that the convergence zone was on the windward side of the ridge with a cloud over the ridgetop. This result was quite different from the previous BIGHILL experiments and could not be explained in this preliminary analysis.

To test if the unusual results might be due to the increased microphysics or taller domain, another run of the high surface heating was made which included the increased microphysics and taller domain. The job was called BIGHTD2 JOB A and produced a cloud in a manner similar to the run without the ice and graupel and in the shorter domain.

2.2 The CCOPE Experiments

The opportunity to run a series of experiments testing the performance of the CSU RAMS model and its microphysics came from the Cooperative Convective Precipitation Experiment (CCOPE) field project. The project collected environmental and cloud physics data on the plains of Montana in the summer of 1981. A series of 14 model runs were made to see how well the model would simulate the storm observed on 19 July 1981 and then test the sensitivity of the model microphysics. The results are presented in two papers

by Banta and Hanson (1985,1986).

Each of the model runs were on a grid of 80 points, 200 meters apart, in the vertical, and 96 points, 250 meters apart, in the horizontal. Each case involved a convergence initialization, represented by a specified updraft in the center of the domain, and an upward warp of surface moisture in the updraft. The winds in the input sounding were reduced by 75 percent to decrease the shear in the u-component of the wind.

Intercomparisons between the observations (including aircraft and radar observations) of the 19 July thunderstorm in Montana and the model were made to verify that the model can adequately simulate the thunderstorm and its environment. These comparisons were made to the run CCOPEH, which specified an initial droplet count of 850 cm^{-3}. Table 1 lists some of the features looked at. The comparisons indicated that the model was capable of simulating many of the features of the storm. The model correctly simulated the early development of the storm, including the rapid growth phase. The model predicted a maximum vertical velocity close to the observed value, and the weakening of the updraft observed after the rapid growth phase was reflected in the model results. The structure of the model cloud agreed with observations with the exception of the cloud base height. The predicted cloud base was 2.9 km MSL, significantly lower than the observed cloud base of 3.9 km MSL. The lower predicted cloud base was probably the result of the initial model sounding, which may have been too moist in the lower levels.

Both the observed cloud and the modeled cloud developed precipitation through the ice phase, with graupel melting and falling out of the cloud as rain. However, aggregates were important, also, in the production of graupel for the simulated cloud, while few aggregates were actually observed in the Montana storm. In turn, the radar echoes from both of the clouds developed similarly. The radar reflectivities from the model simulation, however, did not indicate the rapid decrease in precipitation as was found in the reflectivities of the observed cloud. The continued presence of ice and aggregates within the modeled cloud with graupel and rain falling out of the cloud kept the model reflectivities high.

Parameter	Observation	Model
Cloud base height	3.9 km (+1°C)	2.9 km (+8°C)
max cloud top height (MSL)	10.5 km (-48°C)	10.7 km (-48°C)
width of liquid cloud	6.0 km x 6.0 km	5.5 - 6.0 km
process via ice phase	yes	yes
dominant ice type	graupel	graupel
max liquid water content	2.5 g m^{-3}(4.2 g kg^{-1})	3.3 g m^{-3}(5.0 g kg^{-1})
(mixing ratio)	at 7.0 km MSL	at 6.2 km MSL
	(adiabatic)	(84% adiabatic)
first echo height	6.0 - 6.0 km	4.4 - 6.3 km
max reflectivity	55 dBz	65 dBz
max updraft	10 - 15 m s^{-1}	15.5 m s^{-1}
sudden death of cloud	yes	no

Table 1. Comparisons of Observed and Modeled Cloud

The sensitivity studies involved making several runs using the same initial conditions was changing the concentration of the initial cloud droplets (representing cloud condensation nuclei). Initial cloud droplet concentrations of 200, 500, 600, 700, 850, 1,000, and 10,000 cm^{-3} were tested to find how the concentration affected the microphysical processes in the model cloud. With the small concentrations rain formed before the graupel, indicating that precipitation was mainly the result of autoconversion of cloud droplets to rain drops. With the higher concentrations rain formed, after the graupel had formed indicating that the precipitation was the result of ice processes and that graupel had formed by riming and not by freezing of raindrops. The intermediate cases were 500, 600, and 700 cm^{-3} where rain was initially

produced by warm rain processes but later precipitation was also produced by melting.

For most of the CCOPE model runs, the critical radius of the cloud droplets needed to initiate conversion to raindrops specified was 0.0010 cm. To test the sensitivity of the microphysics to the value of the critical radius, three additional runs were made with the critical radius changed to 0.0011 cm, 0.0012 cm, and 0.0015 cm; all three runs used an initial cloud droplet concentration of 500 cm^{-3}. Increasing the critical radius had the same effect as increasing the initial droplet concentration. Both the 0.0012 cm and the 0.0015 cm cases produced graupel through ice processes with only 500 cm^{-3} for the initial droplet concentration.

Two cases were run without including the aggregation process in the microphysics. These used an initial cloud droplet concentration of 1000 cm^{-3}. When aggregation was turned off, a considerable amount of water ended up as small ice crystals, and the conversion to graupel was very inefficient. This indicates that aggregation is an important mechanism leading to graupel in the RAMS model. In the second case which did not include aggregation, the ice crystal concentration was diagnosed by the Fletcher (1969) expression instead of predicted as in the other runs. This run produced higher graupel amounts than that including aggregation and slightly lower rain amounts without the contribution of melting aggregates.

3. CLOUD AND MESOSCALE MODEL EXPERIMENTS USING THE CSU MODEL

3.1 The CSU Cloud and Mesoscale Model

A copy of the CSU RAMS resides on the computer system at the Air Force Weapons Laboratory (AFWL), Kirtland AFB, NM for use by the Cloud Physics Branch. The model system contains a pre-processor software package to allow the use of various options in the model physics, or to specify two or three space dimensions.

The modular Fortran code for the model is stored in several files on the IBM 4341 at AFWL (which front-ends their Cray I); a list of the modules is given in Appendix 1. Through the use of an exec program called M on the IBM, updates to those files can be made and applied to the files containing the original source code without having to actually change the original source code. That way, each model experiment has its own set of updates (stored in its job file with its data and job control language) and only one copy of the original source code is necessary. The updates are applied to the original source code each time M is used to submit a job to the Cray 1 computer at AFWL.

The model experiments actually run on the Cray. As the model runs, output can be saved on "analysis tape" (not a physical tape) files. The data in these files can be used by an analysis program to produce plots of the model results. Information about the variables and parameters is also stored, as the model is running, on "history tape" files; from this information a model run can be restarted at model times other than zero seconds.

Programs which analyze the output stored on the analysis tape files were also developed at CSU. The analysis programs also are modular and the routines are stored in several files on the IBM (see Appendix 1 for a list of the files). The exec program M is used to make updates, apply the updates, and submit the programs to the Cray.

The following sections describe how different model experiments, which OPHIR Corporation was involved in, were created and how several of the analysis programs were modified to analyze the results of the model studies.

3.2 Model Experiments

3.2.1 KEELEY Jobs

The first work undertaken by OPHIR with the model involved an older version of the model before it was called RAMS. Those jobs were called KEELEYx JOB A, where x is a number, and they were used to gain familiarity with the model. The KEELEY jobs did not represent a particular study. Since the KEELEY jobs involved the older version of the model, the exec program MOD was used to work with the updates and submit the jobs to the Cray instead of the exec program M.

3.2.2 *ATOMIC and EXPLODE Jobs*

The experiment ATOMIC JOB A was an attempt to simulate the effects of a sudden, large temperature perturbation, as might be expected from an atomic explosion, on the atmosphere. The experiment was originally named ATOMIC JOB A and was later called EXPLODE JOB A. It used the older version of the CSU mesoscale model and the MOD exec program. The model was not designed to handle this sort of situation and some initialization problems were encountered. The study was intended to be continued with the installation of RAMS version 2 but an emphasis on other studies in the modeling program of the Cloud Physics Branch prevented the completion of this study.

3.2.3 *BIGHILL Jobs*

The RAMS model was next used to simulate the lee side convergence zone mechanism for thunderstorm initiation. The convergence zone on the lee side of a mountain range had been simulated in a two dimensional experiment, by Dr. Robert Banta of the Cloud Physics Branch, of flow over a small hill without moisture parameters. A larger hill and moisture parameters were introduced to learn if the convergence zone would lead to a cloud and thunderstorm. Several experiments were run to test the sensitivity of the model to a stronger surface heat flux and domain scale convergence. Changes were made to version 2 of the RAMS model to set up these experiments.

3.2.3.1 *BIGHILL*

The first job was named BIGHILL JOB A and it evolved from Dr. Robert Banta's experiment BUMP2D JOB A. Appendix 2 lists the modifications which converted the BUMP2D experiment to BIGHILL JOB A, where the hill was made larger and warm rain processes were included. The history tape names were changed from HBUMPA, HBUMPB, and HBUMPC to HHILL1 through HHILL6. And the analysis tape names were changed from ABUMPA, ABUMPB, and ABUMPC to AHILL1 through AHILL7. The experiment name was BIG HILL UPSLOPE WITH WARM RAIN.

3.2.3.2 *BIGHTD*

The second experiment was called BIGHTD JOB A and it was designed to test the effect of increasing the surface heating to 25° cm s^{-1}. BIGHTD JOB A was copied from BIGHILL JOB A and only the value of the surface heating was changed.

In the subroutine SFCLYR of SURF3 MODEL,

```
WTV=17.
```

was changed to

```
WTV=25.
```

In the DATA, the history files were renamed hbhtd1-hbhtd6 and the analysis files were renamed abhtd1-abhtd7. The experiment name was changed to HEATED BIGHILL - UPSLOPE WITH WARM RAIN.

3.2.3.3 *BIGCNV*

To learn how large scale convergence affects the development of thunderstorms in the lee side convergence zone, a third experiment was run with weak convergence imposed over the domain. This experiment was named BIGCNV JOB A and it was created by making some changes to BIGHILL JOB A.

To add the convergence initialization to BIGCNV, add a file fetch command using the M EXEC routine on the IBM. Fetch the file CONV2 INIT D.

In subroutine INITILZ of DRIVER2 INIT, add the statements

```
      DO 330 K=1,NZP
      WM(K)=WM(K)*WMSCAL
  330 CONTINUE
```

which add a scaling factor to the mean vertical motion imposed in the domain. Add them just after the READ(1,INDAT).

In CNFIG RAMS, the scaling factor WMSCAL was added to the common block /SOUNDG/ in the global STORAGE and to the namelist /INDAT/.

In the DATA, add to the INDAT namelist

```
WM=10.,12.,15.,19.,25.,32.,40.,48.,59.,77.,31*100.,95.,85.
     ,75.,65.,55.,45.,35.,25.,16.,10.,8.,7.,6.,5.,4.,3.,2.
     ,2*1.,5*0.                        ,;MEAN W FOR CONVERGENCE INIT
WMSCAL=0.1                             ,;SCALING FACTOR FOR WM
```

Also change

```
ADJTIM=0.
SPNTIM=0.
```

The history files in this experiment were named hbcnv1-hbcnv6 and the analysis files were named abcnv1-abcnv7. The experiment name was HEATED BIGHILL - WARM RAIN.CONV (WM MAX=10CM/S).

3.2.3.4 BIGINT

The experiment BIGINT JOB A was created from BIGHILL JOB A to see if the dynamics of the westerly winds flowing over the hill were partly responsible for the thunderstorm development. It was initialized with no cold pool in the valley and no surface heat or moisture fluxes were imposed. A taller domain was used and the initial winds increased with height, similar to those observed in South Park, CO on 3 August 1977.

In the DATA, the initial cold pool in the valley was eliminated by changing the input sounding to adiabatic near the surface. So the lowest six temperature values (beginning at the surface) were changed to 16.8, 15.7, 14.6, 13.3, 9.6, and 8.4.

The initial winds were modified to increase with height in the DATA:

```
USNDG=6*0.2,O.4,1.2,1.5,15*2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5
  ,7.0,7.5,8.0,8.5,9.0,9.5,10.0,10.5,11.0,11.5,12.0,12.5,13.0,13.5
  ,14.0,14.5,16*15.0
```

Also in the DATA, IMID was changed by

```
IMID=41
```

In CNFIG RAMS, the number of vertical grid points was extended with

```
.SE NZ=96
```

In the subroutine SFCLYR of SURF3 MODEL, the surface heating was turned off and the moisture flux was eliminated by setting

```
WTV=0.
DQT=0.
```

The history tape files were called hbint1-hbint6 and the analysis tape files were called abint1-abint7. The experiment name was changed to BIGHILL INITIALIZATION - NO FORCING.

3.2.3.5 BIGNOF

BIGNOF JOB A was created from BIGINT JOB A. It used a shorter domain and a wind profile with less wind shear. To reduce the cost of the run, the microphysical options were turned off.

One of the changes made to BIGNOF JOB A was to decrease the number of vertical grid points. In CNFIG RAMS,

```
.SE  NZ=64
```

set the number of grid points to 64. Changes were made to CNFIG RAMS to turn off the microphysical options in the model. The line

```
.AC  D
```

was changed to

```
.EL  M
```

Then the printed output had to be changed to avoid printing the microphysics. The DATA were changed to read

```
NPLT=4                                    ,;4 PLOTS IN ALL
IAA=4*41,IAB=4*60,JOA=4*1,JOB=4*30        ,
```

The wind field was also changed in the DATA:

```
USNDG=6*0.2,0.4,1.2,1.5,10*2.0,2.5,3.0,3.5,4.0,4.2,4.4,4.6,4.8,5.0
   ,5.2,5.4,5.6,5.8,6.0,6.2,6.4,6.6,6.8,7.0,7.2,7.4,7.6,7.8,8.0,8.2
   ,8.4,8.6,8.8,9.0,9.2,9.4,9.6,9.8,13*10.0          ,
```

and IMID was reset to

```
IMID=27
```

To insure that the output was saved when a job reached a time limit, the statement in subroutine MODEL of DRIVER2 MODEL

```
IF(TR.LT.30)
```

was changed to

```
IF(TR.LT.90.)
```

The history files were renamed hbnof1-hbnof6 and the analysis files were renamed abnog1-abnof7. The new experiment name was BIGHILL NO MICROYPHYS- NO FORCING, NZ=64.

3.2.3.6 BIGHILL with a Longer Wind Spin-Up Time

Another study was made with BIGHILL JOB A. The ambient winds in this study were added more slowly, over a period of 30 minutes and the winds were allowed to settle for an additional 30 minutes before the surface heating began. In this case, fourth order advection and an initial cloud droplet concentration of 850 cm^{-3} were used.

In this experiment, the time that the surface heating was begun was changed in subroutine SFCLYR of SURF3 MODEL

TIM1=3600.

In the DATA, the wind spin up time was changed, fourth order advection was turned on, and the initial droplet concentration was changed

TIMSCL=1800.
IAV4=1
CON=850.

The history tape files for this study were named hhilg1-hhilg6 and the analysis tape files were named ahilg1-ahilg7. The experiment name was BIGHILL UPSLOPE W/WARM RAIN, 30X30 SPIN.

3.2.3.7 BIGHTD with a Longer Wind Spin-Up Time

A second experiment was also made with BIGHTD JOB A which included all of the changes made to the second experiment of BIGHILL JOB A (with a longer wind spin-up time). For the second BIGHTD JOB A study, the history files were renamed hbhtl1-hbhtl6 and the analysis files were renamed abhtl1-abhlt7. The experiment name was changed to HTD BIGHILL- W/WARM RAIN, GTR HEAT, 30X30 SPIN.

3.2.3.8 BIGCNV with a Longer Wind Spin-Up Time

Another experiment was conducted with BIGCNV JOB A. This second study included all of the changes that were made to both BIGHILL JOB A and BIGHTD JOB A for a longer wind spin-up time. In addition to those changes, BIGCNV JOB A used an expanded domain with 128 grid points in the vertical and ice and graupel processes were included in the microphysics.

The initial wind sounding was expanded for just the BIGCNV JOB A case, in the DATA

USNDG=6*0.2,0.4,1.2,1.5,120*2.0

Changes were also made to CNFIG2 RAMS. The number of vertical grid points was increased by

.SE NZ=128

Ice and graupel were turned on in the microphysical options but the ice concentration was not predicted

.AC E
.EL G

To correct a problem with the model, the line

ESE NPLMX=6

was added after the line

DSE NPLMX=6 .

The history files were named hbcnl1-hbcnl6 and the analysis files were named abchl1-abcnl7. The experiment name was HTD BIGHILL- W/ICE,128PT,CONV(WM MAX=10 CM/S).

3.2.3.9 BIGHTD2

A third experiment with BIGHTD JOB A was run from the file BIGHTD2 JOB A . BIGHTD2 JOB A was the same as BIGHTD JOB A with a longer wind spin-up plus it has 128 vertical grid points and included the ice and graupel microphysical processes that were added to BIGCNV JOB A (see section 3.2.3.8 'BIGCNV with a Longer Spin-Up Time' for the list of those changes). The history files were called hbht21-hbht26 and the analysis files were called abht21-abht27. The experiment name was HEATED

BIGHILL W/ICE 128PT.

3.2.4 CCOPE

The CCOPE model experiments were designed to compare a model simulation to actual observations of a small thunderstorm. The storm was observered near Miles City, Montana during the CCOPE field experiment in the summer of 1981. In the course of the study, sensitivity tests of the model were made by varying the initial cloud droplet concentration and the critical radius of the droplets at which autoconversion to raindrops began. Also, the influences of including aggregation in the microphysical processes modeled and predicting the ice crystal concentration were tested.

To create the CCOPE model experiments from the previous experiment BIGCNV JOB A (which included domain scale convergence), many modifications were necessary. The CCOPE studies used flat terrain and included both warm and cold cloud microphysical processes. Winds were included in the initial sounding, the convergence was focused at a point, and grid spacing and domain size were changed. Appendix 3 contains the modifications made in creating the CCOPE jobs.

3.2.4.1 CCOPEA

The study CCOPEA JOB A was run with the autoconversion of cloud droplets to rain drops turned off so that the rain could form only as the result of ice processes. The specified initial concentrations of cloud droplets was 850 cm^{-3}. The history file was named hcopa1 and the analysis files were named acopa1 and acopa2. The experiment name was CCOPE 19 JUL 81 - 80X96 REDUCED SHEAR.

3.2.4.2 CCOPEB

CCOPEB JOB A was also run with the autoconversion turned off. The scaling factors WISCAL and WMSCAL were reduced from 0.7 to 0.5 to reduce the strength of the initially specified updraft. The history file was called hcopb1 and the analysis files were called acopb1 and acopb2. The experiment name was CCOPE 19 JUL 81 - REDUCED SHEAR - WISCAL=.5.

3.2.4.3 CCOPEC

The autoconversion process was turned on for the experiment CCOPEC JOB A and WISCAL and WMSCAL were again set to 0.7. The specified initial cloud droplet concentration CON was changed to 1000 cm^{-3}. The history file was named hcopc1 and the analysis files were named acopc1 and acopc2. The experiment name was CCOPE 19 JUL 81 - REDUCED SHEAR - CCN1E3.

3.2.4.4 CCOPED

In the study CCOPED JOB A, the initial cloud droplet concentration was set to 10,000 cm^{-3}. The history file was called hcopd1 and the analysis files were called acopd1 and acopd2. The experiment name was CCOPE 19 JUL 81 - REDUCED SHEAR - CCN1E4.

3.2.4.5 CCOPEE

The initial cloud droplet concentration CON was set to 500 cm^{-3} for the case of CCOPEE JOB A. The history file was named hcope1 and the analysis files were named acope1 and acope2. The experiment name was CCOPE 19 JUL 81 - REDUCED SHEAR - CCN5E2.

3.2.4.6 CCOPEF

The initial cloud droplet concentration was reset to 200 cm^{-3} for CCOPEF JOB A. The history file was called hcopf1 and the analysis files acopf1 and acopf2. The experiment name was CCOPE 19 JUL 81 - REDUCED SHEAR - CCN2E2.

3.2.4.7 CCOPEG

In the experiment CCOPEG JOB A, the initial cloud droplet concentration was reset to 700 cm^{-3}. the history file was named hcopg1 and the analysis files were named acopg1 and acopg2. The experiment name was CCOPE 19 JUL 81 - REDUCED SHEAR - CCN7E2.

3.2.4.8 CCOPEH

In CCOPEH JOB A, 850 cm^{-3} was used as the initial cloud droplet concentration. The name of the history file was hcoph1 and the names of the analysis files were acoph1 and acoph2. The experiment name was CCOPE 19 JUL 81 - REDUCED SHEAR - CCN8.5E2.

3.2.4.9 CCOPEI

The initial cloud droplet concentration in CCOPEI JOB A was 600 cm^{-3}. The history tape was called hcopi1 and the analysis tapes were called acopi1 and acopi2. The experiment name was CCOPE 19 JUL 81 - REDUCED SHEAR - CCN6E2.

3.2.4.10 CCOPEJ

CON, the initial cloud droplet concentration for CCOPEJ JOB A was again 1000 cm^{-3}. In addition the microphysical processes for aggregation were not included for this experiment. The history tape was named hcopj1 and the analysis tapes were named acopj1 and acopj2. The experiment name was CCOPE 19 JUL 81 - NO AGGREGATES - CCN1E3.

3.2.4.11 CCOPEK

The experiment CCOPEK JOB A used an initial cloud droplet concentration of 1000 cm^{-3}. It did not include the microphysical processes for aggregation and it did not predict the ice crystal concentration. The history file was hcopk1 and the analysis files were acopk1 and acopk2. The experiment name was CCOPE 19 JUL 81- NO AG .EL G - CCN1E3.

3.2.4.12 CCOPEL

An initial cloud droplet concentration of 500 cm^{-3} was specified in CCOPEL JOB A. Aggregation processes and the prediction of ice crystal concentrations were turned on for this study. The critical radius of the cloud droplets needed to initiate conversion to raindrops was changed to 0.0012 cm (previous runs used a critical radius of 0.0010 cm). The critical radius is defined in the subroutine MICONST of MICRO2 MODEL by setting the variable

```
RADCR=.0012
```

The history file was called hcopl1 and the analysis files were called acopl1 and acopl2. The experiment name was CCOPE 19 JUL 81 - RADCR=.0012 - CCN5E2.

3.2.4.13 CCOPEM

The experiment CCOPEM JOB A was similar to the experiment CCOPEL JOB A. The only difference between the two is that the critical radius for conversion to raindrops was set to 0.0015 cm in CCOPEM JOB A. The name of the history file was hcopm1 and the names of the analysis files were acopm1 and acopm2. The experiment name was CCOPE 19 JUL 81 - RADCR=.0015 - CCN5E2.

3.2.4.14 CCOPEN

The critical radius was set to 0.0011 in CCOPEN JOB A. All other parameters were set the same as in CCOPEL JOB A. The name of the history file was changed to hcopn1 and the names of the analysis files were changed to acopn1 and acopn2. The experiment name was CCOPE 19 JUL 81 - RADCR=.0011 - CCN5E2.

3.3 Analysis Programs

The analysis programs use software available on the Cray at the National Center for Atmospheric Research (NCAR) to produce plots in NCAR metacode. The NCAR Graphics System software was installed on the Cray 1 at AFWL and a description of the installation procedure is given in section 4 of this report. The analysis programs use a metacode translator to convert NCAR metacode to AFWL metacode. The AFWL metacode can be directed to any device at AFWL (i.e. microfiche plotter or tektronix terminal) to draw the plots.

A description of several analysis jobs is given below. It is important that the file CNFIG2 RAMS (or CNFIG5 RAMS with version 5 of RAMS) be exactly the same in the analysis job for a particular run as it was in the model job itself. Fields are to be plotted, contour limits and contour intervals, and limits on the y-axis are specified in the DATA file for each analysis job. To let the graphics routines set the limits of the y-axis for the integral plots, use the default value of 1.E36 for the top and bottom limits specified in the DATA.

3.3.1 Analysis Jobs for the BIGHILL Experiments

Several analysis jobs were created to plot the results of the BIGHILL experiments. ANL4 JOB A was one of these jobs which analyzed the fields produced by several of the BIGHILL model runs. The other analysis jobs for the BIGHILL experiments were similar with just their CNFIG2 RAMS (or CNFIG5 RAMS) files differing from ANL4 JOB A. ANL4 JOB A evolved from B2DANL JOB A which plotted the results of the bump2d studies. Numerous modifications were made to the DATA file in creating ANL4 JOB A; the most resent version of the DATA file is given in Appendix 4.

Several modifications were made to the model configuration file CNFIGA ANLMDL. In the global CLDSTR,

```
BBBOT(NINTG),TTTOP(NINTG)
```

was added to the common block /INPUTA/ and to the namelist /INPUTA/,

```
BBBOT,TTTOP
```

was added. The value of these variables was specified in the DATA; they determine the value of the top and bottom limits of the Z axis in some of the plots. In the global INTLB, to make a plot of upslope winds AINTLB(3) was defined as

```
AINTLIB(3)=U(KK,II,JJ)
```

and in the data block for INITLB, MIN W was changed to

```
PK UPSLP
```

Background field 28 was redefined to be relative humidity with

```
BKLIB(28)=100.*BKLIB(11)/RSA
```

DQSFC was also added to the common block /INPUTS/ and the namelist /INPUTS/ in CNFIGA ANLMDL so that moisture at the lowest grid point can be modified when plotting the moisture field. An ASSIGN statement was included in subroutine DRIVER of CNFIGA ANLMDL

```
CALL ASSIGN(18,5Htape9,ISPACE)
```

just after the line

```
CALL ANALIO(ISPACE)
```

In Subroutine HISTRD of ANLPK1A ANLMDL, the statement

```
TEM(1,7)=TEM(1,7)+DQSFC
```

was added after

```
TEM(1,5)=TEM(1,5)+DTSFC
```

In order to allow the option of setting the limits on the y-axis of the integral plots, several subroutine calls were added to the subroutine DRAWINT of ANLPK2 ANLMDL. The lines

```
C
C
      CALL ABSETF("Y/MINIMUM.",BBBOT(N))
      CALL ABSETF("Y/MAXIMUM.",TTTOP(N))
C
```

were added just before the call to EZXY.

Also in ANLPK2 ANLMDL in subroutine AVERAGE, the line which begins

```
AINTG(IFIL,N)=AMIN1...
```

was changed to

```
AINTG(IFIL,N)=AMIN1(AINTG(IFIL,N),AINTLB(INFNM(N)),1.0)
```

A statement was added to the JCL, to make sure that all the plots are sent to the microfiche plotter, just before the DISPOSE TAPE99 statement:

```
compact tape99 tape99
```

3.3.2 Analysis Jobs for the CCOPE Experiments

3.3.2.1 COPANL

The analysis job COPANL JOB A was used to plot the results of most of the CCOPE model runs. COPANL JOB A was created from ANL4 JOB A.

Many changes were made to the DATA for ANL4 JOB A so that output which was more relevant to the CCOPE studies could be plotted. The DATA is listed in Appendix 5. It is important to note that the precipitation at specific points should be the last integrals requested in the DATA and that the number of the points for which surface precipitation is requested should be specified in the variable NPCPX in the DATA. It is also important that the first point (IPCPN(1)) be requested first and the second point (IPCPN(2)) be requested second, etc. If another order of the plots is desired change the values (of the x-direction grid points) in the array IPCPN, which is also in the DATA.

Changes were also made to the other files and are listed here.

To display the temperature in degrees C instead of degrees K in the temperature movie cross sections, the background needs to be redefined in the file CNFIGA ANLMDL. After BKLIB(36) is defined, the following lines were added.

```
C     TO PLOT TEMP IN DEGREES C, SUBTRACT 273.16
      BKLIB(9)=BKLIB(9)-273.16
```

To plot the accumulated precipitation at several surface points, NPCPX was added to the global CLDSTR in CNFIGA ANLMDL. It was added to the common block INPUTA and the namelist INPUTA on the lines which also contained NTERMS and ITERMS. The maximum number of plots was increased.

```
.SE NINTG=9
.SE NINTLB=19
```

Lines were added to the INTTLB data array after 8HTOT VAP.

```
*,     8HPCPN(X1)
*,     8HPCPN(X2)
*,     8HPCPN(X3)
*,     8HPCPN(X4)
```

Four values were added to the end of the data array ITYILB.

```
...0,9,9,9,9
```

Several lines were added after AINTLB(15) in the global INTLB.

```
C
    DO(L=1,4)
        N=15+L
        IF(ID.EQ.IPCPN(L).AND.JD.EQ.JPCPN(L))
            IF(K.EQ.1)
                AINTLB(N)=PRECIPA(ID,JD)
            ELSE
                AINTLB(N)=0
            ENDIF
        ENDIF
    ENDDO
```

A few changes were made in the global CLDSTR. A line was added to the common block INPUTA.

```
*,  BBBOT(NINTG),TTTOP(NINTG),IPCPN(4),JPCPN(4)
```

A line was added to the namelist INPUTA.

```
*,  BBBOT,TTTOP,IPCPN,JPCPN
```

And several lines were added after the data list for ITYILB.

```
 DATA     (IPCPN(K),K=1,4)
*/         20,22,24,26
*/
 DATA     (JPCPN(K),K=1,4)
*/         1,1,1,1
*/
```

In the data statement INTTLB, the title PK UPSLOPE was changed to

```
MIN W
```

and AINTLB(3) was reset

```
AINTLB(3)=W(KK,II,JJ)
```

In Subroutine BACKGD of ANLPK1A ANLMDL, change the symbols used to depict ice and aggregates by modifying the calls to PWRITX:

```
IF(RIB(II,JJ,IP1).GE.1.E-5)CALL PWRITX(AA,BB,5H'446',5,1,0,0)
IF(RAGB(II,JJ,IP1).GE.1.E-5)CALL PWRITX(AA,BB,5H'445',5,1,0,0)
```

The following lines were added to subroutine ANALRD of ANLPK2A ANLMDL

```
CCC     NOTE! PRECIP AT SPECIFIC POINTS SHOULD BE LAST INTEGRALS IN DATA
        NST=NUMINT-NPCPX+1
        DO (N=NST,NUMINT)
           NNN=N-NST+1
           AINTG(IFIL,N)=PRECIPA(IPCPN(NNN),JPCPN(NNN))
        ENDDO
```

before the line

```
500     CONTINUE
```

3.3.2.2 ZANL

To aid in the comparison of the model results to observations in the CCOPE studies, plots of radar reflectivity with respect to time and height were added to the analysis program. The version of the analysis program used for this was called ZANL4 JOB A. Many modifications were made to the data which is given in Appendix 6.

In the subroutine REFLECT of ANLPK1A ANLMDL, all instances of the variable ZAG were changed to ZA. The subroutine REFLECT was copied to the file ANLPK2A ANLMDL as REFLECT1 and the first line was changed to

```
   SUBROUTINE REFLECT1(RR,RI,RG,RA,ZR,ZI,ZG,ZA,ZP,RO,T,RM,DGM,ROG,
  *GNI,DI,DMA,ROA,ZRG,ZRGA)
```

After the line

```
F        *+ZA
```

the following lines were added.

```
E        ZRG=ZR+ZG
F        ZRGA=ZRG+ZA
```

And the lines

```
E        IF(ZRG.GT.0.) ZRG=10.*ALOG10(ZRG)
F        IF(ZRGA.GT.0.) ZRGA=10.*ALOG10(ZRGA)
```

were added to the end of the routine.

In CNFIGA ANLMDL the maximum number of time sections was increased to 7 and the timesection library was increased by 7.

```
.SE  MXSCTN=7
.SE  NSCTNLB=30
```

In the global CLDSTR, the following names were added to DATA (TMSTLB(KK),KK=21,NSCNLB)

```
*,       8HDBZ ZR
*,       8HDBZ ZI
*,       8HDBZ ZG
*,       8HDBZ ZA
*,       8HDBZ ZP
```

```
*,      8HDBZ ZRG
*,      8HDBZ ZRGA
```

To DATA (TMSFLB(KK),KK=21,NSCTNLB) seven lines of

```
*,      7HOPF6.2
```

were added. In the DATA (ITYTLB(K),K=1,NSCTNLB),

```
,2,2,2,2,2,2,2
```

was added to the end of the list. In the global TMSLB, the following lines were added after the definition of TMSLIB(23)

```
        TMSLIB(24)=0.0
        TMSLIB(25)=0.0
        TMSLIB(26)=0.0
        TMSLIB(27)=0.0
        TMSLIB(28)=0.0
        TMSLIB(29)=0.0
        TMSLIB(30)=0.0
        RO=1./ALPB(K)
M       CALL MCDIAG(TMSLIB(5),TMSLIB(6),TMSLIB(7),TMSLIB(8),RO
M                 *,VTR,VTI,VTG,VTA,DI,GNI,ROA,THETAA*PIA,PPP,ENA)
M       CALL REFLECT1(TMSLIB(5),TMSLIB(6),TMSLIB(7),TMSLIB(8)
M                 *,TMSLIB(24),TMSLIB(25),TMSLIB(26),TMSLIB(27)
M                 *,TMSLIB(28),RO,THETAA*PIA,RADM,DGM,ROG,GNI,DI
M                 *,DMA,ROA,TMSLIB(29),TMSLIB(30))
```

3.4 Installing a New Version of RAMS on the AFWL Computer System

As new versions of the RAMS model are developed and made available to the Cloud Physics Branch, they need to be installed on the AFWL computer system. This installation involves more than copying the files to an account on the IBM 4341. Because the AFWL Cray uses a different operating system than the NCAR Cray, some changes need to be made to the model code to have it run properly at AFWL.

The installation of version 5 of the CSU RAMS model on the AFWL computer began by creating a new job file called BIG5B JOB A. BIG5B JOB A was copied from the job file BIGHILL JOB A which used version 2 of RAMS. First of all, the calls to fetch version 2 files were all changed to fetch version 5 files (i.e. FETCH CNFIG2 RAMS was changed to FETCH CNFIG5 RAMS) in BIG5B JOB A. Because of the differences between the model codes in the version 2 files and version 5 files, the updates used by BIGHILL JOB A were not sequenced properly to be applied to the version 5 model code used by BIG5B JOB A. The updates from the job file BIGHILL JOB A thus could not be used and were eliminated from BIG5B JOB A. However, the modifications contained in those updates are needed to make the model run correctly. These modifications were made to the new version (new updates were created) by comparing the updates and their position in the version 2 code to the version 5 code and then placing the modifications in the same relative position in the version 5 code.

Some of the modifications which had been made to the version 2 code had been made directly to the files containing the program modules and were not in the updates of the BIGHILL JOB A job file. These changes were also made to BIG5B JOB A. These modifications included changing all of the calls to the routines RELEASE, ASSIGN, ACQUIRE, and DISPOSE in subroutine TAPJCL of TAPE5 UTIL to be calls to RELEASE1, ASSIGN1, ACQUIRE1, and DISPOSE1, which are included in the library RAMSPRT. RELEASE, ASSIGN, ACQUIRE, and DISPOSE are system routines on both the NCAR and AFWL Crays, but they do not perform the same functions or require the same parameters be passed to them on both machines.

In the subroutine DRIVER of DRIVER5 INIT the statement

```
CALL SETIO(IBLK2)
```

was added as the first line in the routine.

A problem with the new code was traced to the subroutine SOILW in SURF5 MODEL and the line

```
D   IF(RGP(2,I,J).GT.0)
```

was changed to

```
E   IF(RGP(2,I,J).GT.0)
```

A new MISCLIB RAMS5 was part of the version 5 package. It contains files which are in the library used by the model. To use the new MISCLIB RAMS5, the library needed to be brought up to date. But before the library was brought up to date, an error in the subroutine PRT2D of MISCLIB RAMS5 was corrected by changing

```
116  FORMAT(I8)
```

to

```
116  FORMAT(I1)
```

Then a new library RAMSPRT was created to replace RAMS99 and it included the new routines of MISCLIB RAMS5. See section 5.6. "Adding to a Library on the Cray" for details on how to do this.

Basically, the same procedure that was used to set up BIG5B JOB A under version 5 was followed when creating an analysis job using the version 2 of the analysis code. The new analysis job 5BANL JOB A was copied from ANLXX JOB A. In 5BANL JOB A, the calls to fetch the file CNVIG2 RAMS, CNFIGA ANLMDL, ANLPK1A ANLMDL, and ANLPK2A ANLMDL were changed to CNFIG5 RAMS, CNFIG2 ANLMDL5, ANLPKA2 ANLMDL5, and ANLPKB2 ANLMDL5. The updates for CNFIG5 RAMS were copied from the updates for CNFIG5 RAMS for BIG5B JOB A. As in the model jobs, the updates from the old version of these files were not sequenced properly to be used for the new version. Most of the modifications in those updates were being added to the new analysis code in the same way that modifications to the model code were added to the version 5 of the model code.

Since there was no revision of the files ANLPKRB ANLMDL and INSERT LIB, the fetch commands of those files were not changed and the updates to those two files were left as before.

Some modifications were made to the analysis jobs which were not part of the updates to the old version of the analysis code. The following lines were added to subroutine DRIVER of CNFIG2 ANLMDL5

```
ISPACE=MAX(IBLK2,LOC(THEEND)-LOC(IFILE))
CALL ANALIO(ISPACE)
CALL ASSIGN(I8,5Htape9,ISPACE)
```

just before the statement

```
IOFFM=1
```

Also the loop beginning with

```
DO(N=1,NTAPS)
```

was commented out.

All read statements in the analysis program needed to be changed. In subroutine DRIVER of CNFIG2 ANLMDL5,

```
READ INFO
```

was changed to

```
READ (5,INFO)
```

In subroutine CLDMVY of ANLPKA2 ANLMDL5

```
READ INPUTS
```

was changed to

```
READ (5,INPUTS)
```

And in subroutine ANALYS of ANLPKB2 ANLMDL5

```
READ INPUTA
```

was changed to

```
READ (5,INPUTA)
```

3.5 Model Output Storage

Each time the RAMS model is run and each time the analysis program is run, an output file was sent to the IBM 4341. That output file was printed and stored for each production run of the model. These hardcopy outputs have been given to the Cloud Physics Branch at AFGL.

The history tape files and analysis tape files produced by the model are normally stored on mass storage. For long term storage, the history and analysis tape files are moved to the directory DEEP on the Cray account 1629. Under the directory DEEP, they are archived (stored with use=a) and there is no storage charge for archived files. Copies of model and analysis job files and copies of libraries are also archived in the directory DEEP along with some of the metacode files produced by the analysis jobs. Copies of the original source code modules are also stored there. Appendix 7 lists the files which are archived in the Directory DEEP; capital letters indicate directory files.

The metacode files of plots made in production runs of the analysis programs were routed to the microfiche plotter at AFWL. Those microfiche were mailed to AFGL and have been given to the Cloud Physics Branch.

4. INSTALLATION OF NCAR GRAPHICS ON THE AFWL CRAY

A set of graphics routines (the Graphics System) has been developed by the Scientific Computing Division at the National Center for Atmospheric Research (NCAR). The Graphics System was designed to be device independent and portable. It is available on the Cray at NCAR, where the CSU RAMS model was developed. The analysis programs written to process the results of the RAMS model use the Graphics System extensively. Therefore, it was advantageous to have the Graphics System available on the Cray at AFWL for analysis of model results.

Purchase of the Graphics System included a magnetic tape (labeled as SUE123) with the graphics routines and some test programs to verify that the routines had been correctly implemented. Hard copy documentation was also included; the "Graphics System Implementors Guide", in particular, is helpful when installing the Graphics System on another computer. A description of the contents of the magnetic tape, including a list of the files, came with the tape and is given in Appendix 8.

The analysis programs at AFWL for the RAMS model do call subroutines from the NCAR Graphics System to produce plots. The files used by the programs are TAPE9, BNCARLIB (the compiled version of the routines in NCARLIB3) and XMETALIB (the controllee name of the previously loaded metacode translator TRNSLATE). The specific steps taken to create these files and install the Graphics System on the AFWL Cray are described below.

The NCAR Graphics System software was loaded from tape onto permanent files on the AFWL Cray mass storage. The file NCARTAPE contains a copy of the contents of tape SUE123. The editor on the Cray will not read past the first END OF FILE and is only capable of editing the first of the 79 files stored in NCARTAPE. Thus NCARTAPE was separated into the 79 files included in the graphics package. The 79 files were then combined into one file NCARPLT1 with no END OF FILE separators between them. NCARPLT1 was created so that the entire package would be under one file name with all of the files capable of being edited. Section 5.7. "Reading a Tape on the Cray" gives the procedure for reading a tape, separating the files, and recombining the files.

4.1 The Support Routines

Of the 79 files supplied on the tape of the NCAR Graphics System Software, there are three files which contain 14 support routines needed by the portable system plot package. The Fortran subroutines are ENCODE, PERROR, ULIBER, and WRITEB in the file SPPRT12F and R1MACH and I1MACH in the file MACHR, both of CRAYLIB. Also the Cray assembly language routines GETCHR, IAND, INIT, IOR, ISHIFT, PACKUM, and SETCHR are in the file SPPRT12C of CRAYLIB. All of the Fortran routines were placed together under the file name TESTRY. The consultants at AFWL recommended that the routines written in Cray assembly language be written in Fortran because modifications made to the system at AFWL make these assembly language routines inoperable. The AFWL computer consultant Pat Simari has converted the routines to the Fortran language. They are stored on the file ROUTINES.

The file TEST12 contains a program TEST12 which tests the 14 routines to make sure they work correctly on the Cray at AFWL. Data statements in the program TEST12 were modified to supply machine dependent information for the Cray. The changes are listed in Appendix 9. The program must be attached to the Fortran routines and compiled:

```
COMBINE TESTLOC TEST12 TESTRY ROUTINES <cr>
CFT I=TESTLOC,B=BCFT <cr>

CFT I=TESTLOC,B=BCFT <cr>
```

where the compiled version is stored in BCFT.

```
LDR B=BCFT <cr>
XBCFT <cr>
```

The results of the test are found in the file OUTPUT.

4.2 Test Plots Using the Support Routines

One of the files on the tape of the Graphics System (TESTPLOT of PORTLIB) contains a program which produces NCAR metacode for two simple plots. A program card was added to the beginning of the file

```
PROGRAM TESTPLOT(INPUT,OUTPUT,TAPE6=OUTPUT,TAPE98)
```

and the file was stored on the Cray as TESTPLOT. TESTPLOT requires some routines which are in the file PLOT88 of PORTLIB and are stored on the CRAY as PLOT88. Two data statements in PLOT88 were changed to set the unit number of the metacode file and the smallest positive real number:

```
DATA MUNIT/98/
DATA SMALL/1.E-2000/
```

TESTPLOT also requires some of the basic Fortran routines in the file TESTRY and the Fortran version of the Cray assembly language routines in the file ROUTINES. PLOT88, TESTRY, and ROUTINES were attached to TESTPLOT with the command

COMBINE TESTPIC TESTPLOT PLOT88 TESTRY ROUTINES <cr>

TESTPIC was compiled with

CFT I=TESTPIC,B=BCFT <cr>

where BCFT is the name of the binary file. The program was loaded with

LDR B=BCFT <cr>

and the loader gave the controllee name XBCFT. The word

XBCFT <cr>

was typed to run the program. After the program was run, the file OUTPUT contained any error messages and the file TAPE98 contained the binary metacode.

To get a hex dump of TAPE98, EDIT was used,

EDIT TAPE98 <cr>
. H 0 10 <cr>

These two commands accessed EDIT and printed out the first 10 (octal) lines. This printout was compared to the hex dump given in the comments of TESTPLOT. They were very similar but there were some differences between the two hex dumps. It was thought that the differences might be due to the way EDIT represents control characters or perhaps the comments in TESTPLOT were missing a few words due to typographical errors. Thus, an attempt was made to translate TAPE98 to AFWL metacode.

4.3 The Metacode Translator

The generic translator program provided by NCAR is in the file MCTRPORT of PORTLIB on the NCAR Graphics System tape. A copy of the file is stored in the file TRNSLATE on the Cray at AFWL. The translator was altered somewhat to suit the Cray and the changes are given in Appendix 10. TRNSLATE needs to have the file ROUTINES attached and the command

COMBINE TRNS TRNSLATE ROUTINES <cr>

was used. Then it was compiled with

CFT I=TRNS,B=BTRNS<cr>

Because TRNSLATE contains calls to the AFWL library METALIB, it is loaded with

LDR LIB=METALIB,B=BTRNS <cr>

The loader returned the controllee name which was used to run the program:

XMETALIB <cr>

The AFWL metacode was stored in TAPE99 and the pictures were displayed on a Tektronix terminal with

```
DIRECT I=TAPE99,DEV=TEKTRNX <cr>
```

(a READY? prompt will appear) and

```
PLT1 <cr>
```

or

```
PLT2 <cr>
```

The two plots displayed on the Tektronix were correct when compared to the plots given in Chapter 2 (Implementing the System Plot Package on a Target Computer) of the NCAR manual "The Graphics System Implementors Guide", except that the "e" in exponential expressions was not printed. This was because on the AFWL Cray the routine ENCODE returns a lower case "e" and an upper case is needed.

4.4 Test Plots with TESTPP

The next step was to try the more complicated test program TESTSPP of PORTLIB. The program is in the file TESTSPP and the program line

```
PROGRAM TESTSPP(INPUT,OUTPUT,TAPE98,TAPE6=OUTPUT)
```

was added. The program TESTSPP also requires subroutines from PLOT88, TESTRY, and ROUTINES. They were attached and the program was compiled, loaded, and run in the same manner as TESTPLOT. Once TESTSPP has run, the file TAPE98 contained the NCAR metacode for 27 plots. The translator was run and the results (in TAPE99) plotted on the Tektronix screen in the same manner as before.

4.5 Assembling the Routines Needed for the RAMS Model Analysis

The next step was to gather together in one file all the routines that would be necessary to create the graphics output expected from the CSU cloud model. This large file was named NCARLIB3 and it contains the AFWL Cray files PLOT88, TESTRY, and ROUTINES. Also included in NCARLIB3 were the CRAYLIB files EZMAP and PWRITX, the ULIB files AUTOGTAPH, CONCOM, CONRAN, CONRAQ, CONRAS, CONTERP, HAFTON, ISOSRF, ISOSRFHR, PWRY, PWRZI, PWRZS, PWRZT, SCROLL, SRFACE, STRMLN, THREED, VELVCT, and WINDOW, and the PORTLIB file ENCD all from the NCAR Graphics System tape. Also a dummy subroutine SDACCESS was added to NCARLIB3 because SDACCESS is a systems routine on the Cray at NCAR but not on the Cray at AFWL:

```
SUBROUTINE SDACCESS(II,JJ)
II=0
RETURN
END
```

A file containing contouring routines was needed for NCARLIB3 along with a file containing dashed line routines. Because there are several files of contouring routines (CONRCQCK of CRAYLIB, CONREC of CRAYLIB, and CONRCSPR of ULIB) which have the same entry points but employ different algorithms for contouring, it was important to include just one of these files in NCARLIB3. In the same way, only one of the files of dash routines (DASHSUPR of CRAYLIB, DASHCHAR of ULIB, DASHLINE of ULIB, and DASHSMTH of ULIB) was included. Thus DASHCHAR of ULIB was chosen to be added to NCARLIB3 along with CONREC of CRAYLIB.

The file PWRITX of CRAYLIB (one of the files included in NCARLIB3) requires a local binary file called TAPE9. The procedure for creating TAPE9 is given in Appendix 11.

4.6 Testing the graphics routines of NCARLIB3

Several routines from TESTLIB on the NCAR Graphics System Tape test the graphics routines of NCARLIB3. To test a few of the graphics routines, the TESTLIB files CONRCSMTH, CONREC, DASHCHAR, HAFTON, and PWRITX were combined into the file TESTLIB. A short program was added to the beginning of TESTLIB to call each of the test subroutines. It is shown in Appendix 12.

Finally, TESTLIB was run using

    CFT I=NCARLIB3,B=BNCARLIB <cr>

to compile NCARLIB3, where BNCARLIB is the binary file, and

    CFT I=TESTLIB,B=BTEST <cr>

to compile TESTLIB. They were loaded with

    LDR B=(BTEST,BNCARLIB) <cr>

and run by typing the controllee name

    XBTEST <cr>

The resulting NCAR metacode for the plots stored in TAPE98 was translated to AFWL metacode with TRNSLATE and stored as TAPE99. The plots of TAPE99 displayed on the tektronix terminal agreed with the examples given in Chapter 9 of the NCAR manual "The Graphics System Implementor's Guide."

5. DESCRIPTIONS OF UTILITIES USED

5.1 Kermit between the AFGL Vax and a Zenith Z-100

With the Kermit utility, files may be sent from the Z-100 to the Vax or downloaded from the Vax (currently a Vax 11/780) to the Z-100. After logging into the Vax in the normal manner, enter the directory where the files are stored or will be stored on the Vax. Exit the terminal emulator utility which is being used on the Z-100 and enter the directory where the files will be added or where the files reside which will be sent to the Vax. Enter the Kermit utility on the Z-100 with the command

    KERMIT <cr>

A prompt of KERMIT-MS> will respond. The baud rate should be set to 9600 bps when transferring the data over the local area network (or 1200 bps when transferring over telephone lines). Type

    SET BAUD 9600 <cr>

to do this, return to terminal mode with

    CO <cr>

The Vax will be waiting for a command. To enter Kermit on the Vax type

    KERMIT <cr>

and it will respond with a KERMIT-32> prompt. If the files are in binary code, type the command

    SET FILE TYPE BINARY <cr>

Put Vax Kermit into server mode with the command

SERVER <cr>

and return to the Z-100 Kermit with

<cntrl>-] C

At this point a file may be transferred either from the Vax to the Z-100 or from the Z-100 to the Vax:

1. To upload a file from the Z-100 to the Vax, type:

   SEND filename <cr>

   The status of the file transfer will appear on the screen and the file will be placed in the current working directory on the Vax. When the file has been transferred to the Vax, the prompt KERMIT-MS> will appear and another file can be sent.

2. To download a file from the Vax to the Z-100, type:

   GET filename <cr>

   to the KERMIT-MS> prompt. The file transfer status will be shown on the screen and the KERMIT-MS> prompt will re-appear when the transfer is complete. More files can then be downloaded.

When all the files have been transferred, return to Vax Kermit with

CO <cr>
<cntrl>-Y
<cntrl>-Y

and the KERMIT-32> prompt will re-appear. The Kermit utility on the Vax can be exited by typing

EX <cr>

to the KERMIT-32> prompt. Next, exit the Z-100 Kermit utility by returning to the Z-100 Kermit with

<cntrl>-] C

and typing

EX <cr>

to the KERMIT-MS> prompt. A terminal emulator utility on the Z-100 can then be used to reconnect to the current process on the Vax. For more information, see AFGL Technical Memorandum by D. Keith Roberts.

5.2 Saving Screen Ouput to a File on the Z-100

5.2.1 Saving a Text File with ZSTEM

While in the ZSTEM utility on the Zenith Z-100, text that appears on the screen can also be saved to a file on the Zenith hard disk. The ZSTEM utility is called by typing

ZSTEM <cr>

on the Z-100. The current version of ZSTEM is configured correctly for communications over the AFGL local area network. However, some changes to the configuration setup will be needed before using it with a modem. To change the configuration, hit

<help>

and a ZSTEM? prompt will appear in the bottom left corner of the screen. Type

CONFIGURE <cr>

and a question will appear at the bottom of the screen. Respond with

R

for remote.

Answer the questions to set up the correct baud rate, parity, echo, etc. The current setting is given in parentheses after the question. A

<cr>

will keep the current setting. When all the questions have been answered, a ZSTEM? prompt will remain in the bottom left corner. A

<cr>

will begin terminal emulation. Communications with the host computer can then begin.

To save the output to the screen on the Z-100 hard disk, hit

<help>

and the ZSTEM? prompt will again appear in the lower left corner. Type

DISK <cr>

and answer the question with

W

(write to a file). It will ask for a filename so type

filename <cr>

to tell it the name of the file the output will be stored in. Another

<cr>

will return to terminal emulation mode and the session will be recorded in the Z-100 buffer. When the buffer is full, it will write to the disk. It may be necessary to hit

<cr>

to continue after writing to the disk (or if noise over the telephone lines interrupts the output to the screen). To stop saving to the file, again hit

<help>
DISK <cr>

This time type

C

to close and store the file on the hard disk. Return to the terminal session with a

<cr>

5.2.2 Saving a Graphics File with Flexitek

This section explains how to bring plots of model output from the Cray at AFWL to the Zenith Z-100. From the metacode file created by the analysis program and stored on the AFWL Cray, plots can be directed to a tektronix terminal. The Flexitek utility, available on the Zenith Z-100, emulates a tektronix terminal. This utility is called by the command

TEK <cr>

on the Zenith. The menu will appear on the screen, and the baud rate should be set to 1200 bps.

<f3>

<f3>

will change the baud rate to 1200 bps.

<f8>

<f8>

puts it into terminal emulation mode.

The Cray command

DIRECT I=filename,DEV=TEKTRNX <cr>

sends the metacode to a tektronix terminal. The figure to be drawn is selected by typing

PLTxx

to the READY prompt, where xx is the plot number. To have the screen output saved to a file on the Z-100 hard disk, return to the Flexitek menu with

<f1> ,

activate the disk capture with

<f7> ,

type the name of the file on the Z-100 that the plot will be saved in

filename <cr> ,

and continue as a tektronix terminal with

<f8>

Then hit

<cr>

for the Cray. The plot will be drawn on the Z-100 screen and will be saved in the Z-100 buffer at the same time. When the buffer is full, it will write to the disk. It may be necessary to hit

<cr>

to have the plot continue after writing to the disk (or if noise over the telephone lines interrupts the plot to stop). The READY prompt will appear when the plot is finished.

After returning to the Flexitek menu with

<f1> ,

the remaining contents of the buffer will be written to the file on the disk and the file will be closed with

<f9>

When all the plots have been saved in this manner,

<f10>

will exit the Flexitek program. The graphics files stored on the Z-100 can be sent to the Vax using the Kermit utility or to a pen plotter.

5.3 Laser Prints of Graphics Output

5.3.1 Cray Procedure to Print on the IBM 3800 AT AFWL

Metacode files created by analysis programs and stored on the AFWL Cray can be sent to a laser printer at AFWL (IBM 3800) and mailed to AFGL. The laser graphics output is possible only for AFWL META files like those produced by the analysis programs.

Two page formats are available, I3 and I2. The normal listing page format is device name "I3" and the image is 10.933 inches horizontal (x-axis) by 7.5 inches vertical (y-axis). The document format which is burst and trimmed 8.5 by 11 inch paper, is device name "I2" and the image is a 8.266 by 10.0 inches.

The Cray commands to send a META file to the laser printer are:

```
XDIRECT I=filename,O=tempfilename,DEV=devname <cr>
DISPOSE DN=tempfilename,MF=I1,DC=PR,DIR=GRAPHI3,FID=lyc,WAIT <cr>
```

where filename is the name of the META file, tempfilename is the name of a temporary file, devname is the page format device name (either I3 or I2) and lyc is a code which directs the operators where to mail the printed output (currently the code is LYC for AFGL's Cloud Physics Branch). In the XDIRECT command, "I=filename," is not needed if the META file is named TAPE1 and "O=tempfilename," is not needed if the temporary file name is TAPE2.

5.4 AFGL Vax procedure for laser graphics

Plots of model output from the Cray at AFWL can be printed on the laser printer for the Vax at AFGL. The plots are first saved on the hard disk of the Zenith Z-100 using the Flexitek utility (see section 2.2) and later are sent to the Vax using the Kermit utility (see section 2.1). The Vax command to send a graphics file to the Vax laser printer is

LASERTEK filename <cr>

where filename is the name of the tektronix graphics file to be plotted. It was found that a <cr><lf> (carriage return, line feed) in the file will cause the laser printer to go to another page. To avoid this, a simple program was written (and is included below) to read the file, remove the <cr><lf>'s, and write the new code to another file PLTXX.DAT. The input for the program is

filename <cr>
13 10 <cr>

where 13 and 10 are the ascii codes for <cr> and <lf>, respectively.

```
      implicit integer (a-z)
      character*80 filename
      character*510 buffer
      read(5,100) filename
100   format(a80)
      accept*,i1,i2
      open(unit=1,name=filename,type='old',readonly,recordsize=510)
      open(unit=2,name='pltxx.dat',type='new',carriagecontrol='none')
1     read(1,200,end=2) nb,buffer
      rec = rec + 1
200   format(q,a510)
      do i=1,nb-1
         if (buffer(i:i+1) .eq. char(i1)//char(i2) ) then
         buffer(i:i+1) = char(0)//char(0)
         type*,'rec,i ',rec,i
         end if
      end do
      lines = nb/80
      extra = nb - lines*80
      do i = 1, lines
      write(2,300) buffer(80*(i-1)+1:80*i)
      end do
      if (extra .gt. 0 ) then
      write(2,400) buffer(80*lines+1:nb)
      end if
300   format(a80)
400   format(a<extra>)
      goto 1
2     end
```

5.5 Archiving Files on Cray Mass Storage

Files are stored at AFWL using their mass storage device. There are no storage costs for those files stored with use=a, that is archived. There is a charge to retrieve an archived file from mass storage. The

command to store a file on archive is the STORE command with use=a specified,

MASS STORE USE=A filename <cr>

Archived files are retrieved in the same way as the other files stored on mass storage

MASS GET filename <cr>

See the Cray File System (CFS) documentation in the AFWL manuals for more information on mass storage.

5.6 Adding to a Library on the Cray

To add new routines or replace old routines in a library on the Cray, the new code must first be sent to the Cray from the IBM 4341 using the command

CTSTORE filename filetype filemode

and the file will be called filename in the Cray local file space. Before compiling, the preprocessor will need to be applied. The preprocessor is called pp2gt and is stored on mass storage. It requires that the input file be named pp1fil and the preprocessed code, which is its output, is named ftnfil.

SWITCH filename PP1FIL <cr>
PP2GT <cr>

Then ftnfil can be compiled with

CFT I=filename,B=bfilename <cr>

where filename is the name of the file which contains the new code and bfilename is the name of the binary file which will contain the compiled code. The old library oldlib is brought off of mass storage with

MASS GET oldlib

and

BUILD OL=oldlib,B=(bfilename) <cr>

will update oldlib. Use

MASS STORE oldlib <cr>

to save oldlib on mass storage. Or use

SWITCH oldlib newlib
MASS STORE newlib

to change the name and save it as a new library called newlib. More information on the BUILD command can be found in the AFWL documentation manuals.

5.7 Reading a Tape on the Cray

It is possible to read a tape on CTSS by using the cosmos job SIS. SIS must be a local file and no local files should have the name TAPE1 or TAPE2 when SIS is running. The command is

COSMOS I=SIS WITH STOCI rb nf vsn <cr>

where STOCI means "stranger to local", rb represents the maximum record blocking size (rb should be no greater than 48), nf represents the number of files to be read, and vsn is the name of the tape to be read. With this command all files read from the tape are stored in a local file called TAPE2.

To break up TAPE2 into separate files, use the program OUTFILE. Program OUTFILE should also be local and it requires that the input file be named TAPE1:

DESTROY TAPE1 <cr>
SWITCH TAPE2 TAPE1 <cr>

Then run the program:

TRIXGL O <esc> OUTFILE <cr>

a "." will appear as a prompt

RUN <cr>
END <cr>

Each of the files in TAPE1 is now a separate local file. They are named DISK1 through DISKnf.

To combine these files into one large file called bigfile use the command:

COMBINE bigfile DISK1 DISK2 DISK3 DISK4 <cr>

If the files being combined are large enough, time will run out before the combing is finished. The time limit on COMBINE can be reset by typing

<ctrl>-E tl=10 <cr>

right after the <cr> of the COMBINE command. This, however, should not be done while using the ZSTEM utility on the Zenith Z-100 because <ctrl>-E terminates the program.

5.8 The Autosum Utility on the Cray

Autosum is a program available on CTSS which is an interactive data inquiry tool for retrieving information from the AFWL Integrated Computer Center utilization data bases generated by system utilities. The data is stored in a file under the directory root /AUTOLOG. Files for each month of the year have the path name /AUTOLOG/CYyr/mon/node. Data for the current and preceding month are available under the pathname /AUTOLOG/ODD or EVEN/machid (EVEN for even-number month, ODD for odd-number month).

Once the /AUTOLOG file is brought from mass storage under the name NATIVE, AUTOSUM can be called. The program will ask for selection options (the criteria used by AUTOSUM to retrieve just the information requested), break options (the criteria specifying how the data will be broken down for the output), and display options (the criteria specifying the variables which will be printed out). The results are written to the file OUTPUT and can also be printed on the screen. The run identification is just the title given to a run so that it can be distinguished from other runs in OUTPUT. For a list of options available or more information see the AFWL AUTOSUM manual.

Appendix 13 gives a sample session with AUTOSUM. The results of this example of the January 1984 costs show the service units used, the cp hours used, the cost incurred, the duration used, and the priority used under the charge code of 0000XXXX (where XXXX is the LYC charge code) on CTSS. They are listed for each user number by date and shift.

6. CUMOD MODEL STUDY

A sensitivity study of a warm cumulus cloud model with detailed microphysics (Silverman and Glass, 1973) was performed using the Cyber CYBER 750 and VAX-780.

The study included the model's response to variations in cloud radius, duration of buoyant heat pulse, and number density of condensation nuclei. The results of this study were presented in Scientific Report No. 2 (Eckhardt, 1985).

7. SNOW-TWO SNOWSTORM ANALYSIS

Adiabatic cooling due to large-scale lifting usually leads to the formation of clouds and precipitation systems. Small scale updrafts and downdrafts are normally found in clouds and precipitation systems with large-scale ascent. The magnitude of large scale lifting is small and, in general, on the order of several centimeters per second. Two methods are frequently used to compute large-scale vertical velocities from observed upper-air data; these are the adiabatic and the kinematic methods. The first is based on the assumption that changes of state of atmospheric air are adiabatic, and the second depends on the principle of mass continuity.

In this study, the environmental conditions as well as large-scale lifting for two snowstorm cases during SNOW-TWO, 16-17 January and 23-24 January 1984, were analyzed. Since the adiabatic assumption may not be a good assumption during a period of snowstorm activity (due to the effects of latent heat release and convective transports resulting from convective overturning), the kinematic method was used to compute large-scale vertical motion in this study. The large-scale vertical motion computed, as well as environmental conditions, will provide useful information for the proper choice of initial conditions for numerical simulations of cloud development at AFGL.

The two case studies of snowstorms which occurred during SNOW TWO were made using National Weather Service regular 12 hr rawinsonde data, hourly surface data, satellite data, and surface observations and upper-air observations taken at Camp Grayling, MI. The physical processes which were responsible for the clouds and precipitation for both cases appeared to be quite different.

The analyses for the 23-24 January case showed that the region of deep convection was along the axis of the southerly wind maximum. A high pressure center was situated over the east coast. The west-east pressure gradient increased as a trough intensified in the west. As a result, the speed of the meridional wind increased. The strong meridional wind brought in the warm and moist air from the south in the middle and low troposphere resulting in a conditionally unstable atmosphere above 700 mb. Finally, the large-scale lifting triggered the deep convection and the release of potential instability.

For the 16-17 January case, a cold front passed Camp Grayling, MI approximately at 0300 GMT 17 January. Sharp horizontal temperature and equivalent potential temperature gradients were found below 850 mb in the frontal zone. It is important to note that the atmosphere was stable both ahead of and behind the surface front except below the temperature inversion near the ground. After the passage of the surface cold front, a well mixed layer was evident below the inversion which separated the cold arctic continental air below and warm air above. Subsidence was found behind the cold front which confined the moisture below the inversion. The ascending motion of warm moist air over and ahead of the surface cold front apparently initiated and sustained the convection. As the cold front and the cold air behind it moved eastward, the relatively warm and moist air was lifted, became saturated and produced precipitation.

The paper "Circulation Analysis of Two Snowstorms during the SNOW- TWO Program", by Chen (1984) gives more of the details and discussion of the study. It also includes documentation on the computer programs used to perform the work. Program and data files used in the analysis on the AFGL Cyber and VAX computers are contained on the magnetic tapes LYC600 and LYC601, respectively.

8. STRATEX DATA ANALYSIS

Analysis work was conducted on data from the STRATEX experiment. The examined data were taken during flights through and near marine stratiform clouds in the Pacific. While data were available for

several flight dates, only the 13 June 1976 flight was examined in detail. The data are on two tapes filed as LYC506 and LYC507.

Program CNVRT, which resides on the AFGL CYBER computer, was modified to make an ASCII listing of all data channels contained on a STRATEX tape. These data were then transferred to the AFGL VAX via tape. (Since the Hyperchannel is now in place, it would probably be used in future transfers of data.) The transfer data tapes are filed as LYC504 and LYC505

The data file on the VAX had many groups of extraneous colons. The cause of these was some unknown incompatibility between the two computer systems. A program, PTEST, was written to clean these characters from the raw transferred file.

A Program WRITE was written to read the transferred file after cleaning by PTEST, and write the data in packed form to a VAX tape. The tape was written in binary form with 8192 byte blocks and 80 character records. It is filed as LYC503. There was a bad record in the data at time 14:19:20 that was corrected by using program T3LIST before the final write to tape.

Program STRFIL reads channels off of the tape written by WRITE and places them in one dimensional files. The files each contain one channel of data and one value per record and are thus simple time series files. These files were created to simplify plotting with IDL since, at the time, the art of conducting formatted reads within IDL was unknown.

Analysis of the data was started using IDL interactively with the aid of several functions written in IDL. These functions were DENS, which calculated air density, LEVAP, which calculated the latent heat of evaporation, amd THETA, THETAL, and THETAIL, which calculated various potential temperatures.

Another program DIST was written to analyze the statistical distribution of certain observed and calculated variables.

All programs reside on DRA3:{WURMAN.STRATEX.DECODPROG}. Source code and sample plots have been provided to AFGL.

9. UND DATA ANALYSIS

9.1 Introduction

In the summer of 1983 the University of North Dakota recorded meteorological data over Hanscom AFB during several flights with a Cessna Citation II aircraft, while under contract to M.I.T., Lincoln Laboratory. The data were stored in Perkin-Elmer binary on nine track magnetic tapes at 1600 bytes per inch. Copies of these magnetic tapes and their corresponding flight notes were given to the Cloud Physics Branch.

Four sets of data, T1, T24, TCAM and PMS, were recorded during each Citation flight. The T1 probe data were collected at the rate of 0.98304 seconds, 1.0 hertz, or as an average of the T24 probe data which were recorded at the rate of 0.04096 seconds, 24 hertz. The TCAM data are a collection of all the camera data. The PMS data consist of the data collected by particle measurement systems probes. Table 2 outlines the magnetic data tapes which were received from Lincoln Labs.

AFGL does not have access to a Perkin-Elmer computer so an alternative computer had to be used for the data analysis. Problems developed, however, when an effort was made to read the binary aircraft data on either of the available machines, the Cyber and the VAX. This occurred because a Perkin- Elmer computer stores binary data in IBM binary form which is different from from VAX (Digital) and Cyber (CDC) binary code. One solution to this problem is to locate a Perkin-Elmer computer elsewhere and use the computer to create new ASCII data tapes. These ASCII tapes can be read on either the VAX or the Cyber. Another solution is to write a program which would read in the original IBM binary data and rewrite the data into Digital or CDC binary. This method requires rearranging the order of the binary data bits. Both of these methods were used in this project.

LABEL	*DATE*	*FLIGHT*	*FILE'S*	*NO. RECORDS*
LYC517	8/4/83	#2	T1,T24	209,1560
LYC522	6/15/83	#1	T1,T24	272,1425
LYC523	6/15/83	#1	T24 (cont.)	630
LYC525	8/12/83	#2	T1,T24	159,1179
LYC540	8/12/83	#1	T1,T24	265,1506
LYC541	8/12/83	#1	T24 (cont.)	493
LYC515	8/4/83	#1	T1,T24	146,1095

Table 2. UND Flight Data Tapes from MIT, Lincoln Laboratory. The Data are stored in IBM binary form. All of the measurements are in volts. The calibration constants are stored in the corresponding header records.

9.2 Analysis of the 1 Hertz UND Flight Data

9.2.1 The First Effort Reading the 1 Hertz Data

The first effort to read the T1 data was done on a Perkin-Elmer computer at Lincoln Labs. With the assistance of Ms. Barbara Gonsalves, a Lincoln Labs staff member who was familiar with the Perkin-Elmer operating system, one of the T1 data files was rewritten from binary into ASCII. Due to the lack of available computer time, only one of the T1 data files could be rewritten in ASCII. This new T1 data file was stored on magnetic tape number LYC516. This file is 1 Hertz Data recorded August 4, 1983

9.2.1.1 IDL Graphics of the ASCII Data

The ASCII T1 data were analyzed on the VAX-11 780 at AFGL. This set of flight data may be plotted using the IDL, Interactive Data Language, graphics package. It may be done interactively or by running the IDL program, UNDPLOTS.PRO, a copy of which has been provided to AFGL. The program will ask several questions about the plots that will be created. It will ask if you are plotting on a Tektronix or a Regis terminal. Then a list of the probe files will be displayed. You may choose the x and y points to be plotted from this list. These probe data files should exist in your current directory. A sample plot appears below in Figure 1.

If the time data file is not present you may create it by simply running the following program:

```
      OPEN (UNIT=1, STATUS='NEW', RECL=20, FILE='TIME.DAT')
      X = 599.6544
      Y = 599.6544/60
666   FORMAT (F12.6)
C
      WRITE (2,666) Y
      DO 100 JJ = 1, 4076
      X = X + .983040
      Y = X/60
100   CONTINUE
      STOP
      END
```

9.2.2 The Second Effort Reading the 1 Hertz Data from Tape

Subroutines were located in the user library of the AFGL VAX-11/780 which would change IBM binary into Digital binary form.

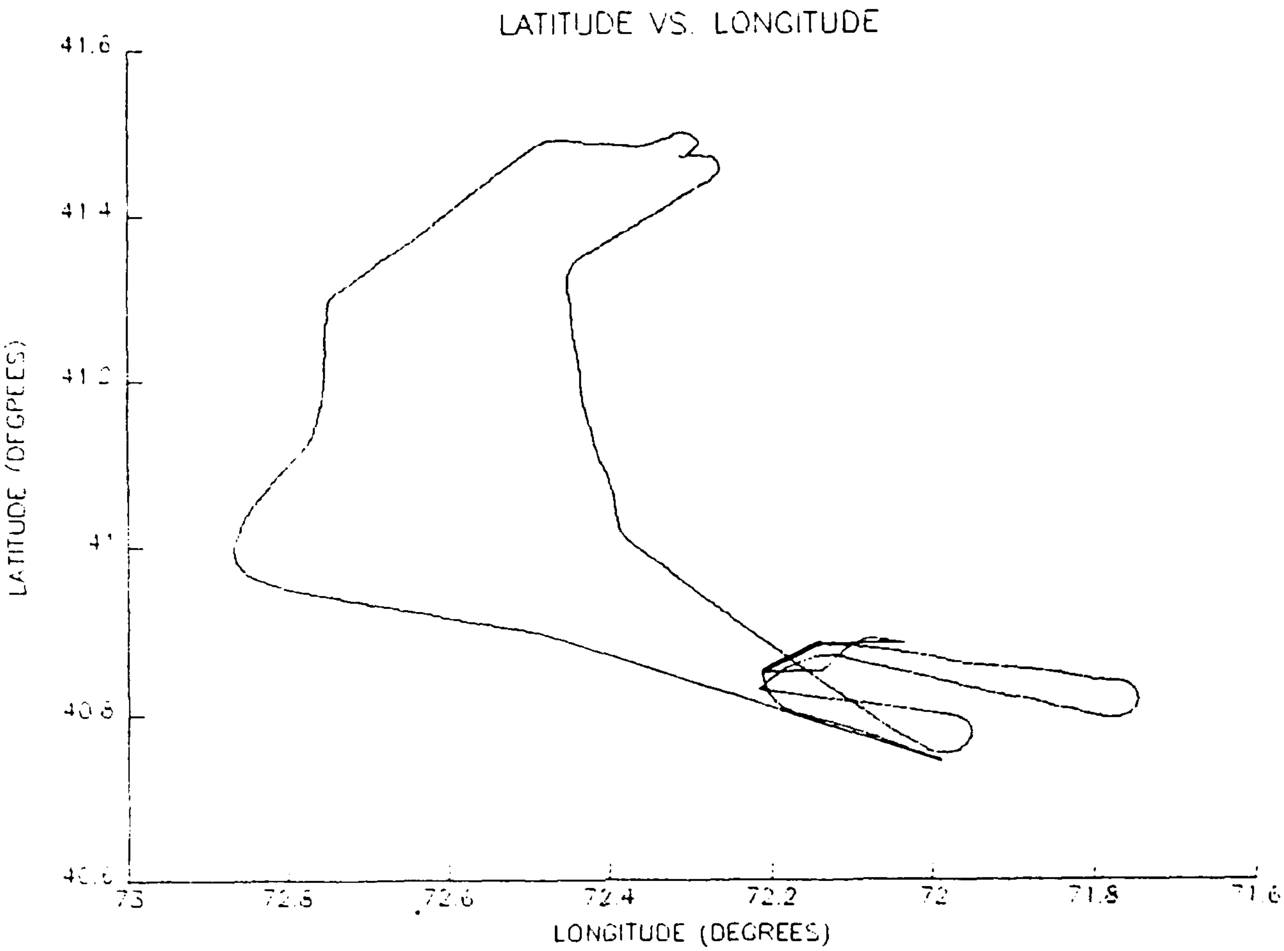


Figure 1. Sample Plot from UNDPLOTS.PRO.

A FORTRAN program was written on the VAX-11/780 which used these three subroutines. Source code has been provided to AFGL. This program, READ1.FOR, reads in the UND Perkin-Elmer binary probe data and rewrites it onto a new tape in Digital binary form. Prior to running the program both the old and the new magnetic tape must be allocated, assigned logical names and mounted. The procedure for mounting the two tapes and executing the program to write on the new Digital tape is the following:

```
$
$ ALLOCATE DEVICE_1              !mount the new VAX tape
$ INITIALIZE/DENSITY=8192 DEVICE_1: TAPE1_LABEL
$ MOUNT DEVICE_1: TAPE1_LABEL LOGICAL_NAME1
$
$ ALLOCATE DEVICE_2              !mount the original tape
$ MOUNT/FOREIGN/DENSITY=32768 DEVICE_2: TAPE2_LABEL LOGICAL_NAME2
$
$ FORTRAN NEWCNVT.FOR
$ FORTRAN READ1.FOR
$ LINK READ1.OBJ,NEWCNVT.OBJ
$ RUN READ1.EXE                  !run the program, READ1.FOR
$
```

The format of the T1 probe data on the new VAX tapes is slightly different than the format of the T1 probe data on the original UND Perkin-Elmer tapes. This new T1 data format is outlined in Appendix 14. Table

LABEL	*DATE*	*FLIGHT*	*# RECORDS*	*DATA FORM*
LYC701	8/4/83	#2	625	VAX BINARY
LYC703	6/15/83	#1	814	VAX BINARY
LYC705	8/12/83	#2	475	VAX BINARY
LYC706	8/12/83	#1	792	VAX BINARY
LYC516	8/4/83	#1	146	ASCII

Table 3. 1 Hertz UND Data Tapes for the VAX. The data is unformatted. The calibration constants must be used to obtain the proper units for each piece of data.

3 outlines the new tapes which were created for the VAX.

After the new UND probe data tape has been created for the VAX the FORTRAN program, ARY.FOR (source code provided to AFGL) may be used. This program will open up to the new VAX tape, read in the probe data and write the data into separate probe data files in disk space on the VAX. These files will be labeled by their tape number and probe number as described below.

FILE NAMES WILL BE...

'TAPE NUMBER' (i.e. 701, 703, 705, or 706) + 'PROBE NUMBER' + .DAT

THE PROBE NUMBERS ARE...

04.........LATITUDE (deg)
05.........LONGITUDE (deg)
08.........WIND DIRECTION (deg)
09.........WIND VELOCITY (knots))
11.........CROSS TRACK DISTANCE (miles)
12.........GROUND SPEED (knots)
13.........TRUE HEADING (deg)
25.........PITOT STATIC NOSE (24 ave. MB)
26.........PITOT STATIC WING (24 ave. MB)
27.........ICE RATE METER (24 ave. volts)
28.........STATIC PRESSURE (24 ave. MB)
29.........ROSEMOUNT TEMPERATURE (24 ave. Celsius)
30.........DEWPOINT TEMPERATURE (24 ave. Celsius)
31.........REVERSE FLOW TEMPERATURE (24 ave. Celsius)
32.........J W LIQUID WATER (24 ave. @ 100 knots)
34.........VERTICAL ACCELERATION (24 ave. m/s2)
39.........ALTITUDE (24 ave. feet)

Prior to executing the program, ARY.FOR, a new VAX T1 data tape must be mounted. The procedure for running the FORTRAN program to create the probe data files is the following:

```
$
$ ALLOCATE DEVICE
$ MOUNT/DENS=8192 DEVICE: TAPE_LABEL LOGICAL_NAME
$
$ FORTRAN ARY.FOR
$ LINK ARY.FOR
$ RUN ARY.EXE
$
```

When the program is done running you may list the probe data files which were created during the programs execution. They will exist in your current directory. For example, if you create probe data files from the T1 data of flight B off tape LYC701, your directory will be the following:

```
$
$ DIR *.

701B04.DAT;1  701B05.DAT;1  701B08.DAT;1  701B09.DAT;1
701B11.DAT;1  701B12.DAT;1  701B13.DAT;1  701B25.DAT;1
701B26.DAT;1  701B27.DAT;1  701B28.DAT;1  701B29.DAT;1
701B30.DAT;1  701B31.DAT;1  701B32.DAT;1  701B34.DAT;1
701B39.DAT;1
```

9.2.2.1 IDL Graphics of the Remaining 1 Hertz UND Flight Data

After these files have been created you may plot them in IDL with the plotting program, LEGS.PRO., provided to AFGL. The program will prompt you for an initial and a final time (minutes) during which the program will select probe data to plot. The program will produce a series of nine plots on a page for each time interval requested. These nine plots are:

LATITUDE/LONGITUDE
STATIC PRESSURE/TIME
ROSEMOUNT TEMPERATURE/TIME
DEWPOINT TEMPERATURE/TIME
VERTICAL ACCELERATION/TIME
JW LIQUID WATER/TIME
POTENTIAL TEMPERATURE/TIME
ALTITUDE/TIME
RELATIVE HUMIDITY/TIME

9.3 Analysis of the 24 Hertz UND Flight Data

9.3.1 Reading the Data Tapes

A FORTRAN program, HZ24.FOR, was written on the VAX-11/780 to convert the data to Digital binary and then write it into separate data probe files on VAX disk space. This program has been provided to AFGL. Prior to running the program one or two (if the T24 data is continued on another tape) of the original UND data tapes must be mounted on one of the VAX tape drives. The procedure for executing the program is the following:

```
$
$ ALLOCATE DEVICE1_NAME:
$ MOUNT/FOREIGN/DENSITY=1600/BLOCKSIZE=32768 DEVICE1_NAME LOGICAL_NAME1
$ ALLOCATE DEVICE2_NAME:
$ MOUNT/FOREIGN/DENSITY=1600/BLOCKSIZE=32768 DEVICE2_NAME LOGICAL_NAME2
$ FORTRAN NEWCNVT.FOR
$ FORTRAN HZ24.FOR
$ LINK HZ24.FOR, NEWCNVT.FOR
$ RUN HZ24.FOR
$
```

After the program is done executing a list of the new data probe files may be found in your current directory, i.e:

```
$
```

$ DIRECTORY *.DAT

UND4005.DAT;1 UND4006.DAT;1 UND4007.DAT;1 UND4008.DAT;1
UND4009.DAT;1 UND4010.DAT;1 UND4011.DAT;1 UND4012.DAT;1
UND4013.DAT;1 UND4014.DAT;1

The name of each data probe file may be broken down into three parts. The first part is "UND40". The "40" represents the last two digits of the first tape's label. The second part is "01", "02", etc. This is the probe number. There are fourteen probes. They are:

01......LATITUDE(DEG)
02......LONGITUDE(DEG)
03......VERTICAL ACCELERATION(M/S**2)
04......ATTACK ANGLE(DEG)
05......PITOT STATIC NOSE
06......PITOT STATIC WING
07......ICE RATE METER
08......STATIC PRESSURE(MB)
09......ROSEMOUNT TEMPERATURE(DEG C)
10......DEW POINT TEMPERATURE(DEG C)
11......REVERSE FLOW TEMPERATURE(DEG C)
12......JW LIQUID WATER
13......VERTICAL ACCELERATION GAINED(M/S**2)
14......ALTITUDE(FT)

The last part, ".DAT", simply identifies the file to be a data file.

After the 14 probe data files were stored on disk space from a flight they were copied to a VAX tape. Then the probe files on disk were divided up into legs using the program 20.FOR, provided to AFGL. The data are stored on the following tapes:

LABEL	*DATE*	*FLIGHT*
LYC544	8/4/83	#2
LYC712	8/12/83	#2
LYC714	6/15/83	#1
LYC716	8/12/83	#1

Table 4. 24 Hertz UND Flight Data. The data is arranged into fourteen data probe files. Each piece of data is formatted and in the correct units.

LABEL	*DATE*	*FLIGHT*	*# LEGS*	*# FILES*
LYC545	8/4/83	#2	2	28
LYC713	8/12/83	#2	1	14
LYC715	6/15/83	#1	4	56
LYC717	8/12/83	#1	4	56

Table 5. 24 Hertz UND Fight Data Divided into Legs. The data is arranged into fourteen data probe files for each leg segment. Each piece of data is formatted and in the correct units.

10. PROGRAMS FOR THE ANALYSIS OF PMS-2D PARTICLE IMAGE DATA

10.1 Program KN2UTIL

The Cloud Physics Branch had requested analysis of PMS-2D data recorded 2 March 1983 on the NASA Convair 990 instrumented aircraft. The data tape (KNE-476) was a standard nine-track PMS-2D particle image tape containing two types of records. The long data records represent 2D particle image slices. One dimension, the columns, is represented by the 32 diode array of the 2D PMS Knollenberg device, while the second dimension represents time. The slow data records occur at ten second intervals and contain VCO and analog information.

Program KN2UTIL is designed to display the information on the PMS-2D tapes in various formats. On the first run on KN2UTIL on the raw PMS-2D data a summary listing of all long and short records was obtained. The summary output is useful in that it provides a means of focusing further analysis to only those records which correspond to the specific time intervals of concern.

10.2 Program KNOLL2D

The program KNOLL2D was originally designed to generate particle concentration tables based upon parameters recorded by the data acquisition systems on board each of the MC-130 aircraft formerly operated by AFGL. In early 1983, AFGL mounted three PMS (Knollenberg) probes on the NASA Convair 990 aircraft to obtain in-cloud measurements of ice and liquid water. Prior to installation aboard the CV-990, some circuitry changes were made to each of the two 2D probes and the PMS Data Acquisition System (DAS). These were necessitated by the much higher true air speed (TAS) attained by the CV-990 as compared to the MC-130 aircraft. The CV-990 typically operated at a TAS of approximately 200 m s^{-1} (versus 100 m s^{-1} for the MC-130). Consequently, the resolution and the size range of the 2D probes were modified. Original specifications for the probes were a 25 μm resolution and a range of 25-800 μm for the cloud probe, and a resolution of 200 μm with a range of 200-6400 μm for the precipitation probe. As flown aboard th CV-990, these specifications were modified to 60 μm (60-1800 μm) and 240 μm (240-7200 μm) respectively.

In order to reduce the data collected by the CV-990 several modifications to program KNOLL2D had to be implemented. Additional changes were also made to KNOLL2D in order to reduce data from a particular flight in which there existed a large amount of "noise" in the TAS information recorded by the DAS. All of the modifications made on program KNOLL2D are described in the following sections. A listing of these changes appears in Appendix 17.

Program KNOLL2D utilizes VCO calibration coefficients in order to convert "counts" (as recorded by the data acquisition system) to true air speed. Two sets of data recorded 25 and 26 January 1983 on the NASA Convair 990 were used to calculate the proper VCO calibration coefficients for true air speed. The first data consist of true airspeed and ADDAS voltage adjusted to counts. The second data set was recorded by the AFGL 2D DAS system which was installed on the Convair 990 to monitor each of the two PMS 2D probes. True airspeed was plotted against counts and linear regression equations calculated for each of the data sets.

When true air speed information recorded on the ADDAS system during an operational flight was compared to both of the calculated linear regression equations, all were found to agree within a range of 50 knots. The actual calibration coefficients were determined by calculating an average of the two regression lines. The result is the linear equation:

$$Y = 99.5 + 0.04*X \quad .$$

The new VCO calibration coefficients must be inserted in card 3 of the data cards used when running program KNOLL2D. Card 3 is labeled $VCOEF and is a namelist card for VCO calibrations.

It was necessary to modify program KNOLL2D to correctly interpret the changes made to the resolution and size range of the 2D probes. As mentioned above the circuitry modifications were made to the 2D probes and the DAS in order to compensate for the greater air speed at which the CV-990 recorded data. KNOLL2D uses TAS as a fundamental variable in all calculations involving number concentrations and sample volume.

When the 2D probes were modified to operate on the CV-990, the width of each of the photo sensitive diodes were increased. Program KNOLL2D utilizes diode width in several calculations including the determination of depth of field (DOF) for the cloud and precipitation probe. Number concentrations are correctly calculated in KNOLL2D by the addition of the proper diode dimensions and DOF equations. These modifications were made in two locations within the program. The diode dimensions are corrected in BLOCK DATA and the DOF equations are modified in FUNCTION PSVOL.

The modified version of KNOLL2D was used to reduce data recorded 2 March 1983 The time interval of concern extended from 19:15:00 to 19:42:00. A total of 8 particle type (pass) cards were inserted into the KNOLL2D procedure file. Each pass card consists of a time interval and a code corresponding to the particle type of interest for that particular interval. A list of the time intervals and corresponding particle types for the cloud and precipitation probes appear in Appendix 18. From the number density data output by KNOLL2D, Tektronix plots of number density versus crystal size were produced for each time interval.

10.3 Program TWODEE

The raw PMS-2D tape must first be reduced by a preprocessing program called TWODEE. Program TWODEE is designed to accept the PMS-2D data acquisition tape and convert the bit patterns into discrete particles described by fundamental parameters.TWODEE also has a summary output option which gives the statistics of each data record on the DAS tape.

10.4 Program PMS2D

The program PMS2D was originally designed to facilitate various forms of automated particle typing of two dimensional cloud probe data. The program incorporates 2-D automatic typing algorithms which were developed by Dr. Hunter of ADAPT Service Corporation. Program PMS2D has undergone a series of revisions in order to produce various forms of output.

It was discovered during this reporting period that the most recent version of the program had been purged from the AFGL computer facility several months previous. This particular version of the program allowed the selection of any one of five different output formats. This was accomplished by setting the input variable LR1 to the integer (1-5) corresponding to the type of output desired. In the previous version of the program LR1 had to be set to one, two, or three, but the most recently updated form of the program provided a VCO (short record) and Pass Totals Summary (option 4), or only a Pass Table Summary (option 5).

OPHIR has reinstated the updated version of program PMS2D on the AFGL CDC computer system. This was accomplished by making modifications to the main program and four subroutines of the original version of the program. A listing of the revisions made to the original program are provided in Appendix 19. Examples of input variable listings with each of the two new output formats generated by these modifications appear in Appendices 15 and 16.

11. GEOGRAPHY PLOTTING ROUTINES

11.1 Summary

A program has been written that aids in the plotting and analysis of geographically oriented data.

The program reads digitized terrain elevation data of varying resolutions and plots contour maps, images, or cross-sections on a tektronix compatible device. The elevation data exist in three resolutions, 5 minute, 30 second, and 3 second. The 5 minute resolution data cover North America, Central America, Europe, and some other small regions of the globe. The 30 second data cover the 48 contiguous states, the 3 second data are only available for New England and Colorado.

Other data can be overlaid over this geographical background. The data can be in the form of locations of events, or groups of locations of events. The density of events can be calculated and contoured over the background.

Cross-sections through the geographic background can be extracted and plotted or written to files. The data array of event density or the geographic background can be filtered or smoothed.

The program is menu driven and has many options to facilitate plotting, screen and file management. Documentation for this program, sample plots, and source code have been delivered to AFGL.

11.2 Documentation

The geography plotter can plot contour maps of digitized terrain data or make pretty images of them. It can also plot geographically oriented data. Other functions of the program facilitate screen and file management.

The program is mostly menu driven. The menu prompts sometimes specify the legal commands for that level but many of the prompts are out of data or absent. Some of the menu levels have a help list that can be displayed by entering "H". These help lists are somewhat out of date. Note that all responses to menu commands should be entered in capital letters and that a carriage return is usually not necessary.

This documentation assumes that the reader is fairly familiar with the VAX/VMS DCL. The reader should, at a minimum, understand concepts concerning directories, accounts, files, and logical names. It is also assumed that the reader is familiar with Tektronix terminals and terminology such as dialog area and segments.

Any user should be keenly aware that this program is not a neatly finished product. There are undocumented features that may be tried. There are also bugs that may interfere with the operation documented here. The program is notably "user-unfriendly" in places where the inputting of incorrect data may cause the program to fail fatally. Use of the program will help familiarize the user with some of the quirks and lead to more productive sessions.

PROGRAM INITIALIZATION

The plotting functions of the program are conducted with routines from the plotting library documented elsewhere. When the initialization routine T_BEG is called at the start of the program, it will search for the logical name T_PLOT_BATCH. If it is TRUE then it will search for the logical symbols T_PLOT_FILE and T_PLOT_DEVICE. T_PLOT_FILE should translate to either TT if the plotting output is to be sent to the terminal, or a valid filename specification if the plots are to be sent to a file for later plotting on a terminal or other device. The logical name T_PLOT_DEVICE specifies the type of terminal or device to which the plots are being sent. Some routines use this information, and some do not. To be sure that the plots come out correctly, T_PLOT_DEVICE should translate to one of the valid device types:

4014	for plots on "green screen" terminals and the LN01 laser printer
4107	for plots on these terminals
4115	" "
4510	for plots being sent to a rasterizer
4662	for plots being sent to the pen plotter

If T_PLOT_BATCH does translate to TRUE, then T_BEG will prompt the user for a device class (TT or a file name) and a device type (4104 etc).

Some other logical names must be defined before running the program. If any contouring is to be done, then the logical CONTOURCTL must contain the filename of the contouring control file. If any altitude data is to be read then logical names must point to the files that contain the appropriate data. T5 must point to the file containing the 5 minute data, G5 must point to the file with the 30 second data, and X5 must point to the file containing the 3 second data. Currently all of these files exist in DRA5:[WURMAN.GDAT] as follows:

t5.dat	5 minute data for N. America Europe etc		
geog30se.dat geog30sw.dat	30 second data for w and e US Note that plot boundaries should be multiples of 0.25 degrees here		

3 second data:

7541.4x4	4x4 degree section of New England with SW at 75W 41N		
7142.1x1	1x1 "	"	71W 42N
7143.1x1	1x1 "	"	71W 43N
11037.6x1	6x1 degree section or Colorado with SW corner at 110W 37N		
11038.6X1	6X1 "	"	110W 38N
11039.6X1	6X1 "	"	110W 39N

The menu commands are described below.

TOP LEVEL:

E Exits the program
K Exits the program and kills all "segments"
L Enters altitude array loading routine LOAD
P Enters PLOT routine
D Enters DATA PLOT routine
A Enters DENSITY PLOT routine
Y Enters CROSS routine
M Enters LINE routine
C Enters VAX COMMAND routine
F Enters FILTER routine
S Enters SCREEN routine
H Displays help menu
R Terminates plotting on current device by calling T_END and then calls T_BEG to reinitialize plotting. This is useful when it is desired to multiple files with different plots in each.

SCREEN ROUTINE:

E Exit to TOP LEVEL
D Enter DIALOG routine
A Modify screen plot boundaries. The program asks for new x_offset, y_offset, width and height for plots. The LOAD routine automatically scales so that they are not distorted at their central latitude. This options allows this to be circumvented before plotting. Enter integer

values.

C Modify a color index range. The program asks for a range of colors and then a hue, lightness, and saturation (5 integers total). All color indices are set to the specified values.

T Transform a segment. The program asks for a range of segments (2 integers), and the x and y scaling factors (2 reals), and a rotation angle (real), and a position (2 integers) for a segment transform.

H Displays a help menu

I Reset the segment start number. The program usually starts making segments at number 100 and then increments from there. This option allows the current segment count to be modified. This is useful if plots from different runs are to be displayed simultaneously without segment conflicts.

V Set segment visibility. The program asks for a segment range and a visibility (3 integers) 0=invisible, 1=visible.

K Kill segment. The program asks for a segment range to kill.

P Set segment position. This is a subset of the T option. The program asks for a segment range and a position (4 integers total)

N Force all pending output. If you think that all graphics output has not been sent to the screen, you can force it by using this option.

DIALOG:

L Set the dialog area lines

T Write graphtext to the screen. The program prompts for a location and height (3 integers), a rotation angle (real) a precision (use 2), and a color index (integer), and then the string to be plotted.

S Set dialog characters small

B Set dialog characters big

P Set dialog area position. Program asks for location of lower left of dialog area (2 integers)

I Set dialog area index. Program asks for foreground and background dialog area color indices (2 integers)

C Set dialog area characters. Program asks for number of characters per line (integer)

E Exit to SCREEN

PLOT:

R Draw image of array. Program asks for two integers. The first specifies the method used

to draw the image.

1 Rectangle fills. Quickest but cannot be stored in segment or sent to rasterizer. This is used on the 4115 for photo sessions and for storing on diskette.
2 Panel fills. Slowest but can be put in panels and sent to rasterizer. This is for the 4107 if the picture is to be stored in a segment.
3 Rectangular panels. Quicker than panels, can be sent to rasterizer, but cannot be displayed on the 4107. Either option 3 or 4 should be tested if you are sending plots to the rasterizer and don't need to see them on the 4107.
4 Rectangular panels with boundaries. For some purposes, it is necessary to draw the boundaries, it is slower than option 3

The second integer determines the method for determining the contour interval where the color index changes. 1 specifies that the color and value array in the control file should be used, 2 specifies that the contour interval in the control file should be used.

The routine that does the plotting is T_ARRAY_COLOR. See the documentation for T_CONTOUR for the specifics on the control file and the common blocks etc.

C Draws a contour map using T_CONTOUR
A Plots axes using T_AXES (The axes will be in color index 4)
S Plots slope and aspect maps. This section is a bit archaic.
T Plots strings with array values on screen.

DENSITY:

This routine loads contour maps of the density of events over a geographic area. The events are placed in bins and then the bin number density is contoured. Normalization to get density/per specified area can be done. For smoother and unbiased plots, data can be smeared over a number of small bins. The program prompts for the bin size, the normalization area if desired, and for smearing. The array must still be plotted using PLOT.

LOAD:

This routine loads the gridded altitude arrays. It must be called before any geographic plotting is to be done because it defines the latitude and longitude ranges for subsequent plots.

The program prompts for a data type. If type=3 is specified (3 second) then southwest corner and range of data in the file is asked. The program then asks for the plot size in degrees, and the southwest corner. It then asks whether to read the array or to return to the top level immediately. If a L is entered then the array is loaded, if not, then the plot boundaries are defined control returns to TOP. This is useful to correct mistakes and to load the plot boundaries while skipping the time consuming process of reading the altitude files.

DATA PLOT:

This routine plots individual geographically located data over the plot region.

Option D:

The data is read from files that contain the latitudes and longitudes of the events, one per record in the same format as in the file dataplot.dat which is in DRA5:[WURMAN]. Note that The fields after the latitude and longitude are correction fields which are applied to the original location in the direction specified by the two character direction at the end of the record. The user is prompted for the name of the data file, and the type and color of markers to be plotted.

Option S:

The data is in the format contained in DRA5:[WURMAN]STNLOC.DAT and is plotted along with a 3 character station identifier. The user is prompted for the identifier color and height, offset from central marker, central marker type and color. This is primarily used to plot surface stations, a fairly complete list of which is in STNLOC.DAT

Option U:

Same as option S but the data format is like that in DRA5:[WURMAN]UPLOC.DAT. This is used to plot upper air stations, a fairly complete list of which is in UPLOC.DAT.

CROSS:

This routine extracts cross-sections along specified directions at specified frequencies and writes them to a file or plots them.

FILTER:

This routine is designed to filter glitches from the altitude data but can be used on any field that is loaded. It filters any number that is thresh less than threshn of its immediate neighbors, only if the number is less than threshz. Thresh, threshn, and threshz are prompted for by the routine.

VAX COMMAND:

This lets the user spawn a subprocess and issue one command at the DCL level, or hit a return and stay in the subprocess until logging out and returning to the program.

LINE PLOT:

This routine is similar to DATA PLOT but plots lines of data. The data file has similar format to the ones in DATA PLOT but each group of data records that is to be joined with a line is preceded with a record containing the number of record in the subsequent group. The user is prompted for the color and style of the line and the type of the markers to be placed on the line. The user is prompted for a code to indicate a wind line (with a marker only at the end point), a merger line (with markers at every point), or a panel

filling the polygon defined by the group of points. The panel is filled with the color specified for the boundary line above.

The source code for the geography programs is contained in DRA5:[WURMAN.GPROG]. The executable image is called GTOP and is also contained in that directory.

A sample annotated plotting session follows.

```
*dabuffer 1000
*daline 32
*

$ def geogt dra4:[wurman.geig.test]
%DCL-I-SUPERSEDE, previous value of GEOGT has been superseded
$ def gd dra1:[wurman.gdat]
$ def g5 gd:geog30sw.dat
$ def t5 gd:t5.dat
$ dir geogt:s*.ctl
%DIRECT-E-OPENIN, error opening DRA4:[WURMAN.GEIG.TEST]S*.CTL;* as input
-RMS-E-DNF, directory not found
-SYSTEM-W-NOSUCHFILE, no such file
$ def geogt dra4:[wurman.geog.test]
%DCL-I-SUPERSEDE, previous value of GEOGT has been superseded
$ rec dir
$ dir geogt:s*.ctl

Directory DRA4:[WURMAN.GEOG.TEST]

SG.CTL;3                                  4   13-MAY-1986 20:02
SG07.CTL;5                                4   14-MAY-1986 17:14
SP.CTL;2                                  4   30-MAY-1986 12:55
STRIPDENS.CTL;5                           4   29-MAY-1986 18:18

Total of 4 files, 16 blocks.
$ copy geogt:sg07.ctl dra3:[wurman]
$ def contourctl sg07.ctl
$ r geogt:gtop
T BEG: Direct plots to terminal (T) or file (F) --> T
T BEG: Enter device type for plots ---------------> 4107
T BEG: Plotting initialized:  file = TT   device = 4107

G TOP: 1986-Feb-21:  Map plotting routine
GNEW:TOP: (L,P,D,M,S,p,E,H,K,C,X,Y) ---> L

G LOAD:  Enter data freq 1=5m 2=30s 3=3s-------> 2

G LOAD:  Enter "Y" to change from 4x4 deg map--> Y

Enter values for lon and lat range (F,F)-->.5,.5
Enter SW corner of plot rwlon,rnlat---> 106.,36.
G LOAD:  Enter a "L" to load array--------------> L
G LOAD: Block  200 Begins with  4830   10700
G LOAD: Block  400 Begins with  4645   11700
G LOAD: Block  600 Begins with  4515    9700
G LOAD: Block  800 Begins with  4330   10700
G LOAD: Block 1000 Begins with  4145   11700
G LOAD: Block 1200 Begins with  4015    9700
G LOAD: Block 1400 Begins with  3830   10700
G LOAD: Block 1600 Begins with  3645   11700

G LOAD: Altitude array loaded
GNEW:TOP: (L,P,D,M,S,p,E,H,K,C,X,Y) ---> S

G SCREEN: Enter screen code (C,V,F,T,D,P,N,E)----> A

current values of x offset,y offset,width,height ent new:
100,100,2500,2500
G SCREEN: Enter screen code (C,V,F,T,D,P,N,E)----> F
```

```
GNEW:TOP: (L,P,D,M,S,D,E,H,K,C,X,Y) ---> P

gplot    0.5000000        0.5000000
G PLOT:    Enter code   (F,S,A,R,C)-------------->    R
input model, mode2
1,2
SML,BIG    1755.000        3742.000
TPR:              61            61            4107            1    6.410000
  6.540000
dx,dy     6.500260        6.371050
GNEW:TOP: (L,P,D,M,S,D,E,H,K,C,X,Y) ---> C

Enter DCL command -->
GEOG SUB>> DIR DRA3:[WURMAN.*]838485.DAT

Directory DRA3:[WURMAN.NEW]

838485.DAT;5                          34   22-APR-1986 21:12
838485.DAT;4                          34   16-OCT-1985 00:00

Total of 2 files, 68 blocks.
GEOG SUB>> E SG07.CTL
    1            1This file will contain plot parameters for plotnew:
*9
    9            9CINT        +00100.0000
*S/100/200/
    9            9CINT        +00200.0000
1 substitution
*EXIT
DRA3:[WURMAN]SG07.CTL;8 41 lines
GEOG SUB>> LOG/BRIEF
GNEW:TOP: (L,P,D,M,S,D,E,H,K,C,X,Y) ---> P-JUL-1986 13:21:5?.21

gplot    0.5000000        0.5000000
G PLOT:    Enter code   (F,S,A,R,C)-------------->    R
input model, mode2
1,2
SML,BIG    1755.000        3742.000
TPR:              61            61            4107            1    6.410000
  6.540000
dx,dy     6.500260        6.371050
GNEW:TOP: (L,P,D,M,S,D,E,H,K,C,X,Y) ---> P

gplot    0.5000000        0.5000000
G PLOT:    Enter code   (F,S,A,R,C)-------------->    A
cht xof yof w h:      55.00000              100          100          2000
       4107
Beginning segment            101
test 1          4107
k}ksed test 1         4107
test 1          4107
passed test 1         4107
test 1          4107
passed test 1         4107
GNEW:TOP: (L,P,D,M,S,D,E,H,K,C,X,Y) ---> D

G DATA PLOT: Enter type code (D,S,U) -----------------------> D
G DATA PLOT: Enter the name of data file ---> DRA3:[WURMAN.NEW]838485.DAT
G DATA PLOT: Enter color and number for markers--> 1,2
g data plot:    106.000   36.000  105.576   36.017    1
```

```
g data plot:    106.000    36.000   105.586    36.020    2
g data plot:    106.000    36.000   105.511    36.127    3
q data plot:    106.000    36.000   105.544    36.236    4
c data plot:    106.000    36.000   105.690    36.179    5
g data plot:    106.000    36.000   105.840    36.446    6
g data plot:    106.000    36.000   105.519    36.201    7
G DATA PLOT: Found           7            0            2            10
gdo:test
Beginning segment          102
tmk:              2          1
G DATA PLOT: Enter the name of data file --->
GNEW:TOP: (L,P,D,M,S,p,E,H,K,C,X,Y) ---> D

G DATA PLOT: Enter type code (D,S,U) ---------------------> S
G DATA PLOT: Enter the name of data file ---> GD:STNLOC.DAT
G DATA PLOT: Ent col,hgt,ofx,ofy,num,col fr ids->1,50,15,15,2,1
g data plot:    106.000    36.000   105.683    36.467    1  E23
g data plot:    106.000    36.000   105.583    36.400    2  T
G DATA PLOT: Found           2            0            2            15
gdo:test
Beginning segment          103
tmk:              2          1
G DATA PLOT: Enter the name of data file --->
GNEW:TOP: (L,P,D,M,S,p,E,H,K,C,X,Y) ---> E

T END: Plotting terminated

G TOP:  Exit requested
$
*STAT HCDAAT
HCDAATTRIBUTES..... 20 2 0
*
```

12. PLOTTING LIBRARY

A plotting library has been developed for the AFGL VAX. The library consists of many high and low level FORTRAN and assembly language callable subroutines. This library was documented in AFGL-TR-86-0014. Source code and sample plots and operating procedures have been provided to AFGL.

13. HAMOD ANALYSIS ROUTINES

A time de[illegible] . model of microphysical processes in convective clouds (HAMOD) is being run on the AFGL CYBER by Cloud Physics Branch Personnel. Output from these runs are transferred to the AFGL VAX-780 using the HYPERCHANNEL. Analysis of the output data is done with programs written in IDL, a high level plotting and data analysis software package resident on the VAX-780. Six analysis programs were written. They reside on the VAX-780 in the directory DRA4:[WURMAN.MORT]. Source code and hardcopies of sample runs have been delivered to AFGL. The six programs are described below.

All the programs were written in the IDL language which is described fully in the IDL User's Guide which is available at AFGL, Cloud Physics Branch. The procedures for invoking IDL and running the programs have been detailed in the hardcopies of sample runs previously delivered to AFGL, Cloud Physics Branch. The output from these programs are plots which can be sent to any of a variety of plotting devices. The currently supported devices are:

1. Tektronix 4510 Rasterizer through a Tektronix 4115 or Tektronix 4107 terminal.
2. Tektronix 4662 pen plotter through a Tektronix 4107 terminal.
3. Tektronix 4691 plotter through a Tektronix 4115 terminal.
4. Digital Equipment Corporation VT-240 terminal.
5. Any terminal capable of emulation Tektronix 4010/4014 graphics.
6. NCAR metacode which can be sent, to many devices including a Digital Equipment Corporation LN01-S laser printer.

The programs are described below.

HAMOD_INDIV_SPECTRA: This program makes plots of outside number density versus particle radius (J class). The user can specify whether to plot liquid or ice concentrations and which model levels and times are to be plotted. Twelve plots are made per page and the output from two model runs can be optionally overlaid.

HAMOD_3D_SPECTRA: This program makes 3D "distorted screen" plots of particle number density versus time and J-Class. The user can specify which four model levels are to be plotted. Liquid and ice number densities are plotted side by side on the plot page.

HAMOD_HEIGHT_XSECT: This program makes plots of vertical velocity height cross-sections, liquid water content, temperature, excess temperature, number density of particles, and reflectivity. The user can specify the model times for these plots. The plots are grouped two to a page so there are three pages of plots per requested time. The three groups are:

vertical velocity -- liquid water content
temperature -- excess temperature
number density -- reflectivity

HAMOD_SUMMARY_DATA: This program makes five time history plots of model output. They are:

1. Total liquid water content due to cloud droplets, raindrops, and hail.
2. Cumulative precipitation from rain and from hail at the ground.
3. Cumulative precipitation from rain and from hail at the cloud base.

4. Precipitation intensity from rain and from ice at the ground.

5. Precipitation intensity from rain and from ice at the cloud base.

HAMOD_2D3D_TIMEHIST: This program makes plots of reflectivity, particle number density, and liquid water content versus time and height. The reflectivity and particle number density plots are made separately for liquid, ice, and total water. The liquid water content plots are made separately for cloud water, liquid water, ice, and total. Each plot is made in contour (2D) and "distorted screen" (3D) form. A total of twenty pages of plots are made.

HAMOD_PR: This file contains the subroutine PR which must be linked to whichever HAMOD analysis program is being run. It is called by the programs to initialize plotting, initialize and terminate and plot pages, and terminate plotting. When plotting is initialized, this routine asks the user to specify the destination device for the plots.

14. MELTING LAYER ATTENUATION STUDY ANALYSIS ROUTINES

During the spring of 1986 aircraft flights in the Boston area were made by Colorado International Corporation under contract to AFGL. The purpose was to study the properties of the precipitation melting layer. Several programs were written or modified to produce data listings, time series plots of flight data, and plots of the Particle Measuring System Inc. (PMS) imagery.

Preliminary analysis was conducted on the AFGL VAX-780. Programs used in this analysis reside in the directory DRA4:[WURMAN.CIC]. Source code, operating instructions, and sample plots have been previously delivered to AFGL.

The programs are described below.

CIC_TAPE_COPY: This program makes an exact copy of the flight data tape.

CIC_TAPE_READ: This program reads the flight data tape and produces a 132 column listing of various flight and thermodynamic parameters. Headings are printed at the beginning of each page.

CIC_TAPE_READ_NH: This program is the same as CIC_TAPE_READ except that the headings are eliminated. The main purpose of the listings produced by this routine is to serve as input to CIC_R_P_L2, which is described below.

CIC_R_P_L2: This program is written in IDL, a high level plotting and data analysis software package resident on the AFGL VAX-780. The program produces several time series plots of flight data. The parameters plotted are: altitude, pressure, Rosemount temperature, true airspeed, dew point temperature, reverse flow temperature, q, potential temperature, liquid water content, equivalent potential temperature, 2dc, fwc, 2dp, M, density, con, and dbar. Output is in the form of a metacode file which can be translated and sent to a variety of devices including the Digital Equipment Corporation LN01-S laser printer.

CIC_PMS_P: This program plots imagery from the PMS probes on the aircraft. All PMS records are plotted along with a record count, time, and probe type. Thirty records are displayed per page. Note that this program uses the graphics routines in the plotting library documented elsewhere in this report and must be relinked to the library if modified.

15. REFERENCES

Banta, R. and K. Hanson, 1986: Sensitivity studies on the continentality of a numerically simulated cumulonimbus. *J. Clim. Appl. Meteor.*, accepted for publication.

Banta, R. and K. Hanson, 1985: Moist convection in a mountain boundary layer model, Preprints, Seventh Conference on Numerical Weather Prediction, June 17 - 20, 1985, Montreal, Canada, pp. 384-387.

Chen, Y.L., 1984: Circulation analysis of two snowstorms during the Snow-Two program, Air Force Geophysics Laboratory, Hanscom AFB, MA, AFGL-TR-84-0255, ADA155225.

Cotton, W.R., M.A. Stephens, T. Nehrkorn, and G.J. Tripoli, 1982: The Colorado State University three-dimensional cloud/mesoscale model - 1982. Part II: An ice phase parameterization. *J. Rech. Atmos.*, **16**, 295-320.

Cotton, W.R., G.J. Tripoli, and R.M. Rauber, 1986: Numerical simulation of the effects of varying ice crystal nucleation rates and aggregation processes on orographic snowfall. *J. Clim. Appl. Meteor.*, **25**.

Eckhardt, R., 1985: A sensitivity study of a one-dimensional time-dependent warm cumulus cloud model, Air Force Geophysics Laboratory, Hanscom AFB, MA, AFGL-TR-85-0331.

McArthur, G.R., 1983: Graphics system implementor's guide, NCAR Technical Note NCAR/TN-165+IA, Scientific Computing Division, NCAR, Boulder, CO.

Roberts, D.K., 1986: How to send tektronix plots to the VAX laser printer using a Z-100 computer, AFGL Technical Memorandum, in print.

Silverman, B.A. and M. Glass, 1973: A numerical simulation of warm cumulus clouds. Part 1: Parameterized vs non-parameterized microphysics. *J. Atmos. Sci.*, **30**, 8, 1620-1637.

Wurman, J.M., 1986: A plotting library for Tektronix compatible devices, Air Force Geophysics Laboratory, Hanscom AFB, MA, AFGL-TR-86-0014. ADA173985

APPENDIX 1 - LIST OF PROGRAM MODULES FOR RAMS VERSION 5 STORED ON THE 'D' DISK OF THE IBM 4341

MODEL FILES

ACOPK5	MODEL	D1
CNFIG5	RAMS	D1
CONV5	INIT	D1
CYCL5	MODEL	D1
DRIVER5	INIT	D1
DRIVER5	MODEL	D1
DRYP5	MODEL	D1
HH5	INIT	D1
HYDPK5	MODEL	D1
MICRO5	MODEL	D1
PLTLB2	RAMS5	D1
PLTLIB	RAMMS5	D1
PRPL5	UTIL	D1
RADPK5	MODEL	D1
SIGP5	MODEL	D1
SURF5	MODEL	D1
TAPE5	UTIL	D1
TURBF5	MODEL	D1

ANALYSIS PROGRAM FILES

ANLPKA2	ANLMDL5	D1
ANLPKA3	ANLMDL5	D1
ANLPKB2	ANLMDL5	D1
ANLPKC2	ANLMDL5	D1
ANLPKD2	ANLMDL5	D1
ANLPKE2	ANLMDL5	D1
ANLPKRB	ANLMDL	D1
CNFIG2	ANLMDL5	D1
CNFIG3	ANLMDL5	D1

APPENDIX 2 - MODIFICATIONS MADE TO THE BIGHILL EXPERIMENT

Changes to the JCL:

Change the job name from jn=bump2d to jn=bighill.
Change history tape name from HTERRAZ to HHILLZ.
Change analysis tape name from ATERRAZ to AHILLZ.

Changes to the DATA:

Change history tape names from HBUMPA, HBUMPB, and HBUMPC
to HHILL1 through HHILL6.
Change analysis tape names from ABUMPA, ABUMPB, and ABUMPC
to AHILL1 through AHILL7.
Change the experiment name to

EXPNME=48HBIG HILL UPSLOPE WITH WARM RAIN

Set the Klemp-Lilly lateral boundary condition with

IBND=2

KMID was set to the value of NZP and IMID was set

IMID=19
KMID=65

To section (6) Boundary Conditions

DISTIM=120.
NFPT=23

was added. Dew point temperatures were added to the sounding in the input data. The sounding is listed below.

PRESS	TEMP	DEW POINT
690.0	9.2	6.9
680.0	9.5	6.9
650.0	9.5	-2.4
640.0	8.5	-6.5
630.0	7.0	-8.9
610.0	4.45	-14.8
600.0	3.2	-15.8
550.0	-3.6	-17.3
540.0	-5.0	-17.0
526.0	-7.0	-16.9
510.0	-8.0	-18.1
500.0	-8.3	-20.7
490.0	-9.0	-23.0
480.0	-10.0	-25.9
400.0	-20.0	-38.5
350.0	-27.0	-46.5

300.0	-35.0	-56.9
250.0	-45.0	-66.9
200.0	-55.0	-80.0
155.0	-63.9	-91.9
150.0	-65.0	-95.0
145.0	-65.0	-95.0
140.0	-65.0	-95.0
100.0	-65.0	-95.0
10.0	-65.0	-95.0

This sounding was meant to represent a typical August sounding in South Park, CO.

Also in the input data, the number of plot slabs that were printed in the model output was changed to six:

```
NPLT=6          ;;6 PLOTS IN ALL
IPLTYP=6*1,
IAA=6*1,IAB=6*60,JOA=6*1,JOB=6*64     ;;PLOT SLAB LIMITS
IDPFL=29,44,51,59,70
```

Changes to CNFIG RAMS:

Set the number of horizontal grid points and activate warm rain microphysics with

```
.SE  NX=128
.AC  D
```

In the global STORAGE, add to the common block PFLS

```
RTPFL(NPFLS,NZ)
```

Changes to subroutine ZSDEF of DRIVER2 INIT:

Set

```
BETA=0.35
```

and

```
ZSTOL=350
```

Changes to subroutine METWVE of DRIVER2 MODEL:

Change CALL WINDIN to CALL WINDIN1 in the statement

```
IF(TIME.LE.TIMSCL+.01.AND.TIMSCL.NE.0) CALL WINDIN
```

Changes to subroutine ACOUSTC of ACOUSTC MODEL:

Move

```
      CALL FILLPFL(8HACOUSTC)
```

from the end of the routine to just before the line

```
     .IN    SMLOUT1
```

Changes were made to subroutine SFCLYR in SURF3 MODEL A to add surface moisture flux. Change

```
      DQT=0.0
```

to

```
      DQT=0.001
```

and add

```
      DTH=DTHV-(0.61*THP(2,I,J)*DQT)
```

just after

```
      IF(DQT.NE.0.0)
```

In subroutine PROFLE of PRPL2 UTIL:

Change

```
      NVRBS=6
```

to

```
      NVRBS=7
```

Change

```
      DATA NAME1/8HZ,8HU,8HTHETA,8HKZ,CM2/S
     D,8HDP/DX,8HNET BOUY
     //
```

to

```
      DATA NAME1/8HZ,8HU,8HTHETA,8HKZ,CM2/S
     D,8HDP/DX,8HNET BOUY,8HRT
     //
```

Change

```
      PRINT 1003,ZPNT,UTMP,THPFL(I1,K),AKMPFL(I1,K),DPDXPFL(I1,K)
     ,,BOUYNET(I1,K)
```

to

```
      PRINT 1003,ZPNT,UTMP,THPFL(I1,K),AKMPFL(I1,K),DPDXPFL(I1,K)
     ,,BOUTYNET(I1,K),RTPFL(I1,K)
```

In subroutine FILLPFL of PRPL2 UTIL:

Add

RTPFL(I1,K)=RTC(K,I,J)

just after the line

UPFL(I1,K)=UC(K,I,J)

APPENDIX 3 - MODIFICATIONS MADE TO THE CCOPE EXPERIMENT

In the DATA, the following parameters were reset.

MINIT is the flag used to turn on a moisture perturbation

MINIT=1

To determine the perturbation size and location within the domain

KMID=17
IMID=49
RAD=3.5E5

The horizontal winds were no longer specified but were in the input sounding, so

USNDG=65*0.0
VSNDG=65*0.0

The grid spacing was changed to

DELTAX=.250
DELTAY=1.00
DELTAZ=.200

The latitude for Miles City, Montana was used to set

RLAT=46.5

The top boundary condition was changed to Orlanski type with

IBND=3
DISTIM=0
NFPT=0

The mixing coefficient parameter and the Klemp-Wilhelmson phase speed were reset with

DKR=.25
CPHAS=0

To read the wind data from the initial sounding,

IWSRC=-1
KMEAN1=2
KMEAN2=-5
IUVFLG=0
UMEAN=200
VMEAN=500

were set. Some changes were made to the printed output options by specifying

IPT=49,49,49,70,70,70
KPT=2,10,20,2,10,20
IAA=6*40,IAB=6*59,JOA=6*1,JOB=6*30

```
IDPFL=29,39,49,59,70
```

Temperature and moisture fluxes were added

```
WTVSFC=20.        ,;SURFACE LAYER TMPFLUX
DRTSFC=-.001      ,;SURFACE LAYER MIXING RATIO JUMP(-FOR UNSTBL)
```

just after the parameter DRTCON. For the convergence initialization, parameters were set as

```
SPNTIM=300.
ADJTIM=600.
```

and some were added after WMSCAL

```
WINIT=0.,10.,23.,40.,56.,71.,85.,
  .98.,7*100.,98.,85.,71.,56.,40.,
  .23.,10.,0.      ,;PERTURBATION W FOR CONVERGENCE INIT
WISCAL=0.7        ,;SCALING FACTOR FOR WINIT
```

The following sounding of pressure, temperature, dew point, wind direction and wind speed was put in the initial data.

PRESS	TEMP	DEWPT	DIR	SPD
930.	25.0	17.8	70.	3.0
910.	24.1	10.7	65.	3.5
900.	24.4	6.9	65.	3.7
850.	20.0	7.2	33.	3.8
800.	15.9	4.8	310.	7.0
750.	11.5	3.5	293.	9.1
700.	7.0	2.8	285.	9.4
665.	3.1	-3.3	295.	11.5
637.	1.2	-10.2	300.	13.0
620.	-0.9	-14.9	302.	13.4
590.	-4.7	-12.1	299.	13.3
573.	-6.0	-12.0	305.	10.8
558.	-6.9	-16.1	315.	10.2
545.	-8.3	-24.7	317.	9.8
522.	-10.2	-21.9	310.	9.1
510.	-11.3	-37.3	307.	10.7
500.	-12.4	-37.5	306.	11.0
480.	-14.2	-38.5	305.	11.6
470.	-15.7	-34.8	305.	11.5
465.	-16.2	-42.0	306.	11.2
440.	-19.9	-33.0	307.	12.5
425.	-20.6	-33.5	303.	14.4
410.	-22.6	-37.6	297.	15.8
400.	-23.4	-29.0	295.	15.3
350.	-30.6	-36.4	294.	11.0
300.	-39.6	-44.6	270.	8.3
245.	-51.4	-81.4	248.	5.2
225.	-51.1	-81.1	258.	11.3
200.	-47.5	-77.5	270.	11.4
165.	-49.0	-79.0	258.	16.1

135.	-55.4	-85.4	270.	15.0
100.	-59.6	-89.6	240.	11.0
10.	-59.6	-89.6	240.	11.0
0.	-99.9	-99.9	-99.99	-99.99

In CNFIG RAMS, the coordinate transformation was not needed for flat terrain and domain size and microphysical tracers were set:

```
.AC Z
```

was changed to

```
.AC Y
```

The statement

```
ESE NPLMX=6
```

was added after

```
DSE NPLMX=6
```

```
.AC D
```

was changed to

```
.AC E
```

Also NX and NZ were specified

```
.SE NX=40
.SE NZ=32
```

The scaling factor WISCAL was added to the common block /SOUNDG/ in the global STORAGE and to the namelist /INDAT/.

In the file DRIVER5 INIT, an imposed updraft scaling factors was used in the subroutine INITLZ

```
WINIT(K)=WINIT(K)*WISCAL
```

just after the line

```
WM(K)=WM(K)*WMSCAL
```

In the subroutine ZSDEF, a 'Z' was placed in the first column of each of the lines creating a 2-d asymmetric hill and in the section printing out the hill (i.e. PRINT 98 and PRINT 99).

In subroutine INITST of HH2 INIT, the calculation of the U and V components of the winds using the wind speed and direction given in the input sounding was corrected.

```
PI180=ATAN(1.0)/90.
```

was changed to

```
PI180=ATAN(1.0)/45.
```

and

```
UMOMS(NS)=SPD*SIN(PI180*DIR)
VMOMS(NS)=SPD*COS(PI180*DIR)
```

was changed to

```
UMOMS(NS)=-SPD*SIN(PI180*DIR)
VMOMS(NS)=-SPD*COS(PI180*DIR)
```

Also the statement

```
    CALL ZSDEF
```

was changed to

```
Z   CALL ZSDEF
```

The line

```
   IF(IUVFLG.NE.O)
```

was changed to

```
   IF(IUVFLG.EQ.0)
```

to correct an error in the code. For this study, the sounding winds were reduced by 75% by changing

```
        SPD=VMOMS(NS)
```

to

```
CCC     TEMPORARILY REDUCE WINDS TO .25
        SPD=VMOMS(NS)*0.25
```

In subroutine SOUND, the section of code which was added to eliminate the upstream cold pool in the BUMP2D case study was commented out (a 'C' was placed in the first column of each line in that section).

In the file CONV5 INIT, the line

```
IMIDX=NX/2
```

was changed to

```
IMIDX=IMID
```

in both of the subroutines BOMB and BOMB1. The line

```
KTOP=KMID
```

was added just before the line

```
IMIDX=IMID
```

in the subroutine PERTURB.

In the subroutine SFCLYR of SURF5 MODEL,

```
WTV=17.
```

was changed to

```
WTV=WTVSFC
```

and

```
DQT=DRTSFC
```

was added after the statement

```
DQT=-.001
```

In subroutine WINDIN1 of DRYP2 MODEL, the calls to TRNCL2 were modified to read

```
CALL TRNCL2(VCTR10,Z,ZV(I,J),ZTOP,VCTR25,VCTR26,VCTR27,NZ)
```

and

```
CALL TRNCL2(VCTR11,Z,ZV(I,J),ZTOP,VCTR25,VCTR26,VCTR27,NZ)
```

APPENDIX 4 - THE DATA FILE FOR ANL4 JOB A

```
SINFO
IANFLG=1                                 ,;DO ANALYSIS RUN
NMVPL=2                                  ,;NO OF MOVIE CROSS SECTIONS
MAXANF=50                                ,;NO. FILES PER TAPE
S
SINPUTA
                        :INPUT FOR MODEL ANALYSIS PROGRAM

IFIL1=1,IFIL2=21                         ,;LIMITS ON TAPE
I1=5,I2=126,;WINDOW IN X DIRECTION
J1=1,J2=1                                ,;WINDOW IN Y DIRECTION
K1=1,K2=48                               ,;WINDOW IN Z DIRECTION
N3D=0                                    ,;NUMBER OF 3D PLOTS

                        :TIME CROSS-SECTION INFORMATION

NSCTN=2                                  ,;NUMBER OF TIME SECTIONS

                        :TIME CROSS-SECTION 1(MAX W)

ISCTN(1)=1                               ,;NUMBER OF TIME SECTION
TMSCNT(1,1)=0.                           ,;LOWER CONTOUR LIMIT
TMSCNT(2,1)=1000.                        ,;UPPER CONTOUR LIMIT
TMSCNT(3,1)=40.                          ,;CONTOUR INTERVAL

                        :TIME CROSS-SECTION 2(MIN W)

ISCTN(2)=2                               ,;NUMBER OF TIME SECTION
TMSCNT(1,2)=-1000.                       ,;LOWER CONTOUR LIMIT
TMSCNT(2,2)=0.                           ,;UPPER CONTOUR LIMIT
TMSCNT(3,2)=40.                          ,;CONTOUR INTERVAL

                        :TIME CROSS-SECTION 3(AVG W)

ISCTN(3)=3                               ,;NUMBER OF TIME SECTION
TMSCNT(1,3)=-1000.                       ,;LOWER CONTOUR LIMIT
TMSCNT(2,3)=1000.                        ,;UPPER CONTOUR LIMIT
TMSCNT(3,3)=50.                          ,;CONTOUR INTERVAL

                        :INTEGRALS

NUMINT=2                                 ,;NUMBER OF INTEGRALS

INFNM(1)=2                               ,;MAX W OVER DOMAIN
BBBOT(1)=0.                              ,;MIN Y-AXIS VALUE
TTTOP(1)=800.                            ,;MAX Y-AXIS VALUE

INFNM(2)=4                               ,;KINETIC ENERGY
BBBOT(2)=1.E36                           ,;MIN Y-AXIS VALUE
TTTOP(2)=1.E36                           ,;MAX Y-AXIS VALUE

INFNM(3)=3                               ,;MIN U OVER DOMAIN
```

```
BBBOT(3)=-500.                          ,;MIN Y-AXIS VALUE
TTTOP(3)=0.                             ,;MAX Y-AXIS VALUE

$

$INPUTS
                    :INPUT FOR MOVIE CROSS SECTION 1

ITITLE=25H                              ,;CROSSECTION TITLE
IFIL1=01,IFIL2=21                       ,;LIMITS ON TAPE
NUMSLB=1                                ,;INDEX OF ANAL SLAB
IXS=1                                   ,;1=X/Z 2=Y/Z 3=X/Y
IA1=20,IA2=99                           ,;WINDOW IN X DIRECTION
JO1=1,JO2=44                            ,;WINDOW IN Y DIRECTION
FCTRS=3.                                ,;VERTICAL EXAGGERATION
IGRDLOC=0                               ,;ARROWS ONLY AT GRID PTS =1
DT=60.,FREQ=300.                        ,;TIME STEP AND FRAME FREQ
SPDMX=300.                              ,;SPEED OF 1 GRID VECTOR
NUMBK=7                                 ,;NUMBER OF DIFFERENT BACKS
DENSTY=0.5                              ,;DENSITY OF ARROWS
MOVIE=1                                 ,;FRAMES CLOSE
IHOLD=0                                 ,;NO LEADER MOVIE

                    :BACKGROUND 1 INFO (W)

IBKFLG(1)=3                             ,;FIELD DEFINITION
MICRO(1)=0,IVCTFG(1)=1,ICLDB(1)=0       ,;EXTRA INFORMATION FLAGS
BKCNTR(1,1)=-1000.                      ,;LOWER CONTOUR LIMIT
BKCNTR(2,1)=+1000.                      ,;UPPER CONTOUR LIMIT
BKCNTR(3,1)=50.                         ,;CONTOUR INTERVAL

                    :BACKGROUND 2 INFO (THETA)

DTSFC=+1.1                              ,;ADD TO SFC TMP
IBKFLG( 2)=6                            ,;FIELD DEFINITION
MICRO( 2)=0,IVCTFG( 2)=1,ICLDB( 2)=1    ,;EXTRA INFORMATION FLAGS
BKCNTR(1, 2)=310.                       ,;LODPER CONTOUR LIMIT
BKCNTR(2, 2)=330.                       ,;UPPER CONTOUR LIMIT
BKCNTR(3, 2)=0.5                        ,;CONTOUR INTERVAL

                    :BACKGROUND 3 INFO (P')

IBKFLG(3)=5                             ,;FIELD DEFINITION
MICRO(3)=0,IVCTFG(3)=1,ICLDB(3)=0       ,;EXTRA INFORMATION FLAGS
BKCNTR(1,3)=-1000.                      ,;LODPER CONTOUR LIMIT
BKCNTR(2,3)=+1000.                      ,;UPPER CONTOUR LIMIT
BKCNTR(3,3)=50.                         ,;CONTOUR INTERVAL

                    :BACKGROUND 4 INFO (UC)

IBKFLG(4)=2                             ,;FIELD DEFINITION
MICRO(4)=0,IVCTFG(4)=0,ICLDB(4)=0       ,;EXTRA INFORMATION FLAGS
BKCNTR(1,4)=-1000.                      ,;LOWER CONTOUR LIMIT
BKCNTR(2,4)=+1000.                      ,;UPPER CONTOUR LIMIT
```

```
BKCNTR(3,4)=50.                              ;;CONTOUR INTERVAL

                          ;BACKGROUND 5 INFO  (DP/DX)

IBKFLG(5)=29                                 ;;FIELD DEFINITION
MICRO(5)=0,IVCTFG(5)=1,ICLDB(5)=0            ;;EXTRA INFORMATION FLAGS
BKCNTR(1,5)=-1.E-3                           ;;LOWER CONTOUR LIMIT
BKCNTR(2,5)=+1.E-3                           ;;UPPER CONTOUR LIMIT
BKCNTR(3,5)=4.E-4                            ;;CONTOUR INTERVAL

                          ;BACKGROUND 6 INFO  (RT)

DQSFC=+2.E-3                                 ;;ADD TO SFC RT
IBKFLG(6)=12                                 ;;FIELD DEFINITION
MICRO(6)=0,IVCTFG(6)=1,ICLDB(6)=1            ;;EXTRA INFORMATION FLAGS
BKCNTR(1,6)=0.                               ;;LOWER CONTOUR LIMIT
BKCNTR(2,6)=20.E-3                           ;;UPPER CONTOUR LIMIT
BKCNTR(3,6)=1.E-3                            ;;CONTOUR INTERVAL

                          ;BACKGROUND 7 INFO  (RH)

IBKFLG(7)=28                                 ;;FIELD DEFINITION
MICRO(7)=1,IVCTFG(7)=1,ICLDB(7)=1            ;;EXTRA INFORMATION FLAGS
BKCNTR(1,7)=0.                               ;;LOWER CONTOUR LIMIT
BKCNTR(2,7)=100.                             ;;UPPER CONTOUR LIMIT
BKCNTR(3,7)=20.                              ;;CONTOUR INTERVAL

$

$INPUTS
                          ;INPUT FOR MOVIE CROSS SECTION 2

ITITLE=25H                                   ;;CROSSECTION TITLE
IFIL1=01,IFIL2=50                            ;;LIMITS ON TAPE
NUMSLB=1                                     ;;INDEX OF ANAL SLAB
IXS=1                                        ;;1=X/Z 2=Y/Z 3=X/Y
IA1=48,IA2=127                               ;;WINDOW IN X DIRECTION
JO1=1,JO2=44                                 ;;WINDOW IN Y DIRECTION
FCTRS=3.                                     ;;VERTICAL EXAGGERATION
IGRDLOC=0                                    ;;ARROWS ONLY AT GRID PTS =1
DT=60.,FREQ=300.                             ;;TIME STEP AND FRAME FREQ
SPDMX=300.                                   ;;SPEED OF 1 GRID VECTOR
NUMBK=7                                      ;;NUMBER OF DIFFERENT BACKS
DENSTY=0.5                                   ;;DENSITY OF ARROWS
MOVIE=1                                      ;;FRAMES CLOSE
IHOLD=0                                      ;;NO LEADER MOVIE

                          ;BACKGROUND 1 INFO  (W)

IBKFLG(1)=3                                  ;;FIELD DEFINITION
MICRO(1)=0,IVCTFG(1)=1,ICLDB(1)=0            ;;EXTRA INFORMATION FLAGS
BKCNTR(1,1)=-1000.                           ;;LOWER CONTOUR LIMIT
BKCNTR(2,1)=+1000.                           ;;UPPER CONTOUR LIMIT
BKCNTR(3,1)=50.                              ;;CONTOUR INTERVAL
```

```
                              :BACKGROUND 2 INFO (THETA)

DTSFC=+1.1                                  ,;ADD TO SFC TMP
IBKFLG( 2)=6                                ,;FIELD DEFINITION
MICRO( 2)=0,IVCTFG( 2)=1,ICLDB( 2)=1        ,;EXTRA INFORMATION FLAGS
BKCNTR(1, 2)=310.                           ,;LOWER CONTOUR LIMIT
BKCNTR(2, 2)=330.                           ,;UPPER CONTOUR LIMIT
BKCNTR(3, 2)=0.5                            ,;CONTOUR INTERVAL

                              :BACKGROUND 3 INFO (P')

IBKFLG(3)=5                                 ,;FIELD DEFINITION
MICRO(3)=0,IVCTFG(3)=1,ICLDB(3)=0           ,;EXTRA INFORMATION FLAGS
BKCNTR(1,3)=-1000.                          ,;LOWER CONTOUR LIMIT
BKCNTR(2,3)=+1000.                          ,;UPPER CONTOUR LIMIT
BKCNTR(3,3)=50.                             ,;CONTOUR INTERVAL

                              :BACKGROUND 4 INFO (UC)

IBKFLG(4)=2                                 ,;FIELD DEFINITION
MICRO(4)=0,IVCTFG(4)=0,ICLDB(4)=0           ,;EXTRA INFORMATION FLAGS
BKCNTR(1,4)=-1000.                          ,;LOWER CONTOUR LIMIT
BKCNTR(2,4)=+1000.                          ,;UPPER CONTOUR LIMIT
BKCNTR(3,4)=50.                             ,;CONTOUR INTERVAL

                              :BACKGROUND 5 INFO (DP/DX)

IBKFLG(5)=29                                ,;FIELD DEFINITION
MICRO(5)=0,IVCTFG(5)=1,ICLDB(5)=0           ,;EXTRA INFORMATION FLAGS
BKCNTR(1,5)=-1.E-3                          ,;LOWER CONTOUR LIMIT
BKCNTR(2,5)=+1.E-3                          ,;UPPER CONTOUR LIMIT
BKCNTR(3,5)=4.E-4                           ,;CONTOUR INTERVAL

                              :BACKGROUND 6 INFO (RT)

DQSFC=+2.E-3                                ,;ADD TO SFC RT
IBKFLG(6)=12                                ,;FIELD DEFINITION
MICRO(6)=0,IVCTFG(6)=1,ICLDB(6)=1           ,;EXTRA INFORMATION FLAGS
BKCNTR(1,6)=0.                              ,;LOWER CONTOUR LIMIT
BKCNTR(2,6)=20.E-3                          ,;UPPER CONTOUR LIMIT
BKCNTR(3,6)=1.E-3                           ,;CONTOUR INTERVAL

                              :BACKGROUND 7 INFO (RH)

IBKFLG(7)=28                                ,;FIELD DEFINITION
MICRO(7)=1,IVCTFG(7)=1,ICLDB(7)=1           ,;EXTRA INFORMATION FLAGS
BKCNTR(1,7)=0.                              ,;LOWER CONTOUR LIMIT
BKCNTR(2,7)=100.                            ,;UPPER CONTOUR LIMIT
BKCNTR(3,7)=20.                             ,;CONTOUR INTERVAL
$
:EOF
\EOD
```

APPENDIX 5 - DATA FILE FOR COPANL JOB A

```
SINFO
IANFLG=1                                  ,;DO ANALYSIS RUN
NMVPL=1                                   ,;NO OF MOVIE CROSS SECTIONS
MAXANF=60                                 ,;NO. FILES PER TAPE
S
SINPUTA
                    :INPUT FOR MODEL ANALYSIS PROGRAM

IFIL1=01,IFIL2=60                         ,;LIMITS ON TAPE
I1=3,I2=94                                ,;WINDOW IN X DIRECTION
J1=1,J2=1                                 ,;WINDOW IN Y DIRECTION
K1=1,K2=75                                ,;WINDOW IN Z DIRECTION
N3D=0                                     ,;NUMBER OF 3D PLOTS

                    :TIME CROSS-SECTION INFORMATION

NSCTN=6                                   ,;NUMBER OF TIME SECTIONS

                    :TIME CROSS-SECTION 1(MAX W)

ISCTN(1)=1                                ,;NUMBER OF TIME SECTION
TMSCNT(1,1)=0.                            ,;LOWER CONTOUR LIMIT
TMSCNT(2,1)=2000.                         ,;UPPER CONTOUR LIMIT
TMSCNT(3,1)=200.                          ,;CONTOUR INTERVAL

                    :TIME CROSS-SECTION 2(PEAK RAIN)

ISCTN(2)=5                                ,;NUMBER OF TIME SECTION
TMSCNT(1,2)=0.                            ,;LOWER CONTOUR LIMIT
TMSCNT(2,2)=5.E-3                         ,;UPPER CONTOUR LIMIT
TMSCNT(3,2)=.5E-3                         ,;CONTOUR INTERVAL

                    :TIME CROSS-SECTION 3(PEAK RI)

ISCTN(3)=6                                ,;NUMBER OF TIME SECTION
TMSCNT(1,3)=0.0000                        ,;LOWER CONTOUR LIMIT
TMSCNT(2,3)=5.E-3                         ,;UPPER CONTOUR LIMIT
TMSCNT(3,3)=5.E-4                         ,;CONTOUR INTERVAL

                    :TIME CROSS-SECTION 4(PEAK RG)

ISCTN(4)=7                                ,;NUMBER OF TIME SECTION
TMSCNT(1,4)=0.0                           ,;LOWER CONTOUR LIMIT
TMSCNT(2,4)=5.E-3                         ,;UPPER CONTOUR LIMIT
TMSCNT(3,4)=5.E-4                         ,;CONTOUR INTERVAL

                    :TIME CROSS-SECTION 5(PEAK LIQ)

ISCTN(5)=9                                ,;NUMBER OF TIME SECTION
TMSCNT(1,5)=0.0                           ,;LOWER CONTOUR LIMIT
TMSCNT(2,5)=5.E-3                         ,;UPPER CONTOUR LIMIT
TMSCNT(3,5)=5.E-4                         ,;CONTOUR INTERVAL
```

```
                        ;TIME CROSS-SECTION 6(PEAK RAG)

ISCTN(6)=8                      ,;NUMBER OF TIME SECTION
TMSCNT(1,6)=0.0                 ,;LOWER CONTOUR LIMIT
TMSCNT(2,6)=5.E-3               ,;UPPER CONTOUR LIMIT
TMSCNT(3,6)=5.E-4               ,;CONTOUR INTERVAL

                        ;INTEGRALS

NUMINT=9                        ,;NUMBER OF INTEGRALS
IPCPN=41,51,61,71               ,;X PRECIP POINTS
NPCPX=2                         ,;NUM OF PRECIP PTS USED

INFNM(1)=7                      ,;TOT RC OVER DOMAIN
BBBOT(1)=0.                      ,;MIN Y-AXIS VALUE
TTTOP(1)=5.E8                   ,;MAX Y-AXIS VALUE

INFNM(2)=2                      ,;MAX W OVER DOMAIN
BBBOT(2)=0.                      ,;MIN Y-AXIS VALUE
TTTOP(2)=2000.                  ,;MAX Y-AXIS VALUE

INFNM(3)=8                      ,;TOT RR OVER DOMAIN
BBBOT(3)=0.                      ,;MIN Y-AXIS VALUE
TTTOP(3)=5.E8                   ,;MAX Y-AXIS VALUE

INFNM(4)=9                      ,;TOT RI OVER DOMAIN
BBBOT(4)=0.                      ,;MIN Y-AXIS VALUE
TTTOP(4)=5.E8                   ,;MAX Y-AXIS VALUE

INFNM(5)=10                     ,;TOT RG OVER DOMAIN
BBBOT(5)=0.                      ,;MIN Y-AXIS VALUE
TTTOP(5)=5.E8                   ,;MAX Y-AXIS VALUE

INFNM(6)=1                      ,;SFC PRECIP OVER DOMAIN
BBBOT(6)=0.                      ,;MIN Y-AXIS VALUE
TTTOP(6)=5.E9                   ,;MAX Y-AXIS VALUE

INFNM(7)=11                     ,;TOT RAG OVER DOMAIN
BBBOT(7)=0.                      ,;MIN Y-AXIS VALUE
TTTOP(7)=5.E8                   ,;MAX Y-AXIS VALUE

INFNM(8)=16                     ,;SFC PRECIP AT X1
BBBOT(8)=0.                      ,;MIN Y-AXIS VALUE
TTTOP(8)=2.E0                   ,;MAX Y-AXIS VALUE

INFNM(9)=17                     ,;SFC PRECIP AT X2
BBBOT(9)=0.                      ,;MIN Y-AXIS VALUE
TTTOP(9)=2.E0                   ,;MAX Y-AXIS VALUE

$

$INPUTS
                        ;INPUT FOR MOVIE CROSS SECTION 1
```

```
ITITLE=25H                                  ,;CROSSECTION TITLE
IFIL1=01,IFIL2=60                           ,;LIMITS ON TAPE
NUMSLB=1                                    ,;INDEX OF ANAL SLAB
IXS=1                                       ,;1=X/Z 2=Y/Z 3=X/Y
IA1=3,IA2=94                                ,;WINDOW IN X DIRECTION
JO1=1,JO2=60                                ,;WINDOW IN Y DIRECTION
FCTRS=1.5                                   ,;VERTICAL EXAGGERATION
IGRDLOC=0                                   ,;ARROWS ONLY AT GRID PTS =1
DT=60.,FREQ=300.                            ,;TIME STEP AND FRAME FREQ
SPDMX=500.                                  ,;SPEED OF 1 GRID VECTOR
NUMBK=7                                     ,;NUMBER OF DIFFERENT BACKS
DENSTY=0.5                                  ,;DENSITY OF ARROWS
MOVIE=1                                     ,;FRAMES CLOSE
IHOLD=0                                     ,;NO LEADER MOVIE

                    ;BACKGROUND 1 INFO (W)

IBKFLG(1)=3                                 ,;FIELD DEFINITION
MICRO(1)=0,IVCTFG(1)=1,ICLDB(1)=0           ,;EXTRA INFORMATION FLAGS
BKCNTR(1,1)=-2000.                          ,;LOWER CONTOUR LIMIT
BKCNTR(2,1)=+2000.                          ,;UPPER CONTOUR LIMIT
BKCNTR(3,1)=100.                            ,;CONTOUR INTERVAL

                    ;BACKGROUND 2 INFO (THETA)

DTSFC=+1.1                                  ,;ADD TO SFC TMP
IBKFLG( 2)=6                                ,;FIELD DEFINITION
MICRO( 2)=0,IVCTFG( 2)=1,ICLDB( 2)=1        ,;EXTRA INFORMATION FLAGS
BKCNTR(1, 2)=300.                           ,;LOWER CONTOUR LIMIT
BKCNTR(2, 2)=380.                           ,;UPPER CONTOUR LIMIT
BKCNTR(3, 2)=5.0                            ,;CONTOUR INTERVAL

                    ;BACKGROUND 3 INFO (P')

IBKFLG(3)=5                                 ,;FIELD DEFINITION
MICRO(3)=0,IVCTFG(3)=1,ICLDB(3)=0           ,;EXTRA INFORMATION FLAGS
BKCNTR(1,3)=-2000.                          ,;LOWER CONTOUR LIMIT
BKCNTR(2,3)=+2000.                          ,;UPPER CONTOUR LIMIT
BKCNTR(3,3)=100.                            ,;CONTOUR INTERVAL

                    ;BACKGROUND 4 INFO (TEMPERATURE)

IBKFLG(4)=9                                 ,;FIELD DEFINITION
MICRO(4)=0,IVCTFG(4)=0,ICLDB(4)=1           ,;EXTRA INFORMATION FLAGS
BKCNTR(1,4)=-75.                            ,;LOWER CONTOUR LIMIT
BKCNTR(2,4)=55.                             ,;UPPER CONTOUR LIMIT
BKCNTR(3,4)=5.                              ,;CONTOUR INTERVAL

                    ;BACKGROUND 5 INFO (MICROPHYSICS)

IBKFLG(5)=0,BKTITL(1,5)=12HMICROPHYSICS ,;FIELD DEFINITION
MICRO(5)=1,IVCTFG(5)=0,ICLDB(5)=1       ,;EXTRA INFORMATION FLAGS
BKCNTR(1,5)=0.                  ,;LOWER CONTOUR LIMIT
BKCNTR(2,5)=0.                  ,;UPPER CONTOUR LIMIT
```

```
BKCNTR(3,5)=0.                    ,;CONTOUR INTERVAL

                    :BACKGROUND 6 INFO (LIQUID WATER)

IBKFLG(6)=18                           ,;FIELD DEFINITION
MICRO(6)=0,IVCTFG(6)=0,ICLDB(6)=1      ,;EXTRA INFORMATION FLAGS
BKCNTR(1,6)=0.                         ,;LOWER CONTOUR LIMIT
BKCNTR(2,6)=5.E-3                      ,;UPPER CONTOUR LIMIT
BKCNTR(3,6)=1.E-4                      ,;CONTOUR INTERVAL

                :BACKGROUND 7 INFO (ICE CONTENT MIX RATIO)

IBKFLG(7)=19                           ,;FIELD DEFINITION
MICRO(7)=0,IVCTFG(7)=0,ICLDB(7)=0      ,;EXTRA INFORMATION FLAGS
BKCNTR(1,7)=0.                         ,;LOWER CONTOUR LIMIT
BKCNTR(2,7)=5.E-3                      ,;UPPER CONTOUR LIMIT
BKCNTR(3,7)=5.E-4                      ,;CONTOUR INTERVAL

$
:EOF
\EOD
```

APPENDIX 6 - DATA FILE FOR ZANL4 JOB A

```
$INFO
IANFLG=1                              ,;DO ANALYSIS RUN
NMVPL=0                               ,;NO OF MOVIE CROSS SECTIONS
MAXANF=60                             ,;NO. FILES PER TAPE
$
$INPUTA
                  :INPUT FOR MODEL ANALYSIS PROGRAM

IFIL1=01,IFIL2=60                     ,;LIMITS ON TAPE
I1=3,I2=94                            ,;WINDOW IN X DIRECTION
J1=1,J2=1                             ,;WINDOW IN Y DIRECTION
K1=1,K2=75                            ,;WINDOW IN Z DIRECTION
N3D=0                                 ,;NUMBER OF 3D PLOTS

                  :TIME CROSS-SECTION INFORMATION

NSCTN=7                               ,;NUMBER OF TIME SECTIONS

                  :TIME CROSS-SECTION 1(DBZ ZR)

ISCTN(1)=24                           ,;NUMBER OF TIME SECTION
TMSCNT(1,1)=-5.                       ,;LOWER CONTOUR LIMIT
TMSCNT(2,1)=125.                      ,;UPPER CONTOUR LIMIT
TMSCNT(3,1)=10.                       ,;CONTOUR INTERVAL

                  :TIME CROSS-SECTION 2(DBZ ZI)

ISCTN(2)=25                           ,;NUMBER OF TIME SECTION
TMSCNT(1,2)=-5.                       ,;LOWER CONTOUR LIMIT
TMSCNT(2,2)=125.                      ,;UPPER CONTOUR LIMIT
TMSCNT(3,2)=10.                       ,;CONTOUR INTERVAL

                  :TIME CROSS-SECTION 3(DBZ ZG)

ISCTN(3)=26                           ,;NUMBER OF TIME SECTION
TMSCNT(1,3)=-5.                       ,;LOWER CONTOUR LIMIT
TMSCNT(2,3)=125.                      ,;UPPER CONTOUR LIMIT
TMSCNT(3,3)=10.                       ,;CONTOUR INTERVAL

                  :TIME CROSS-SECTION 4(DBZ ZA)

ISCTN(4)=27                           ,;NUMBER OF TIME SECTION
TMSCNT(1,4)=-5.                       ,;LOWER CONTOUR LIMIT
TMSCNT(2,4)=125.                      ,;UPPER CONTOUR LIMIT
TMSCNT(3,4)=10.                       ,;CONTOUR INTERVAL

                  :TIME CROSS-SECTION 5(DBZ ZR+ZG)

ISCTN(5)=29                           ,;NUMBER OF TIME SECTION
TMSCNT(1,5)=-5.                       ,;LOWER CONTOUR LIMIT
```

```
TMSCNT(2,5)=125.                    ,;UPPER CONTOUR LIMIT
TMSCNT(3,5)=10.                     ,;CONTOUR INTERVAL

              :TIME CROSS-SECTION 6(DBZ ZR+ZG+ZA)

ISCTN(6)=30                         ,;NUMBER OF TIME SECTION
TMSCNT(1,6)=-5.                     ;LOWER CONTOUR LIMIT
TMSCNT(2,6)=125.                    ,;UPPER CONTOUR LIMIT
TMSCNT(3,6)=10.                     ;CONTOUR INTERVAL

              :TIME CROSS-SECTION 7(DBZ ZP)

ISCTN(7)=28                         ,;NUMBER OF TIME SECTION
TMSCNT(1,7)=-5.                     ,;LOWER CONTOUR LIMIT
TMSCNT(2,7)=125.                    ,;UPPER CONTOUR LIMIT
TMSCNT(3,7)=10.                     ,;CONTOUR INTERVAL

              :INTEGRALS

NUMINT=9                            ,;NUMBER OF INTEGRALS
IPCPN=41,51,61,71                   ,;X PRECIP POINTS
NPCPX=2                             ,;NUM OF PRECIP PTS USED

INFNM(1)=7                          ,;TOT RC OVER DOMAIN
BBBOT(1)=0.                          ,;MIN Y-AXIS VALUE
TTTOP(1)=5.E8                        ,;MAX Y-AXIS VALUE

INFNM(2)=2                          ,;MAX W OVER DOMAIN
BBBOT(2)=0.                          ,;MIN Y-AXIS VALUE
TTTOP(2)=2000.                       ,;MAX Y-AXIS VALUE

INFNM(3)=8                          ,;TOT RR OVER DOMAIN
BBBOT(3)=0.                          ,;MIN Y-AXIS VALUE
TTTOP(3)=5.E8                        ,;MAX Y-AXIS VALUE

INFNM(4)=9                          ,;TOT RI OVER DOMAIN
BBBOT(4)=0.                          ,;MIN Y-AXIS VALUE
TTTOP(4)=5.E8                        ,;MAX Y-AXIS VALUE

INFNM(5)=10                         ,;TOT RG OVER DOMAIN
BBBOT(5)=0.                          ,;MIN Y-AXIS VALUE
TTTOP(5)=5.E8                        ,;MAX Y-AXIS VALUE

INFNM(6)=1                          ,;SFC PRECIP OVER DOMAIN
BBBOT(6)=0.                          ,;MIN Y-AXIS VALUE
TTTOP(6)=5.E9                        ,;MAX Y-AXIS VALUE

INFNM(7)=11                         ,;TOT RAG OVER DOMAIN
BBBOT(7)=0.                          ,;MIN Y-AXIS VALUE
TTTOP(7)=5.E8                        ,;MAX Y-AXIS VALUE

INFNM(8)=16                         ,;SFC PRECIP AT X1
BBBOT(8)=0.                          ,;MIN Y-AXIS VALUE
TTTOP(8)=2.E0                        ,;MAX Y-AXIS VALUE
```

```
INFNM(9)=17                                  ,;SFC PRECIP AT X2
BBBOT(9)=0.                                  ,;MIN Y-AXIS VALUE
TTTOP(9)=2.E0                                ,;MAX Y-AXIS VALUE

S

SINPUTS
                        ;INPUT FOR MOVIE CROSS SECTION 1

ITITLE=25H                                   ,;CROSSECTION TITLE
IFIL1=01,IFIL2=03                            ,;LIMITS ON TAPE
NUMSLB=1                                     ,;INDEX OF ANAL SLAB
IXS=1                                        ,;1=X/Z 2=Y/Z 3=X/Y
IA1=3,IA2=94                                 ,;WINDOW IN X DIRECTION
JO1=1,JO2=60                                 ,;WINDOW IN Y DIRECTION
FCTRS=1.5                                    ,;VERTICAL EXAGGERATION
IGRDLOC=0                                    ,;ARROWS ONLY AT GRID PTS =1
DT=60.,FREQ=300.                             ,;TIME STEP AND FRAME FREQ
SPDMX=500.                                   ,;SPEED OF 1 GRID VECTOR
NUMBK=7                                      ,;NUMBER OF DIFFERENT BACKS
DENSTY=0.5                                   ,;DENSITY OF ARROWS
MOVIE=1                                      ,;FRAMES CLOSE
IHOLD=0                                      ,;NO LEADER MOVIE

                        ;BACKGROUND 1 INFO (W)

IBKFLG(1)=3                                  ,;FIELD DEFINITION
MICRO(1)=0,IVCTFG(1)=1,ICLDB(1)=0            ,;EXTRA INFORMATION FLAGS
BKCNTR(1,1)=-2000.                           ,;LOWER CONTOUR LIMIT
BKCNTR(2,1)=+2000                            ,;UPPER CONTOUR LIMIT
BKCNTR(3,1)=100.                             ,;CONTOUR INTERVAL

                        ;BACKGROUND 2 INFO (THETA)

DTSFC=+1.1                                   ,;ADD TO SFC TMP
IBKFLG( 2)=6                                 ,;FIELD DEFINITION
MICRO( 2)=0,IVCTFG( 2)=1,ICLDB( 2)=1         ,;EXTRA INFORMATION FLAGS
BKCNTR(1, 2)=300.                            ,;LOWER CONTOUR LIMIT
BKCNTR(2, 2)=380.                            ,;UPPER CONTOUR LIMIT
BKCNTR(3, 2)=5.0                             ,;CONTOUR INTERVAL

                        ;BACKGROUND 3 INFO (P')

IBKFLG(3)=5                                  ,;FIELD DEFINITION
MICRO(3)=0,IVCTFG(3)=1,ICLDB(3)=0            ,;EXTRA INFORMATION FLAGS
BKCNTR(1,3)=-2000.                           ,;LOWER CONTOUR LIMIT
BKCNTR(2,3)=+2000.                           ,;UPPER CONTOUR LIMIT
BKCNTR(3,3)=100.                             ,;CONTOUR INTERVAL

                        ;BACKGROUND 4 INFO (TEMPERATURE)

IBKFLG(4)=9                                  ,;FIELD DEFINITION
MICRO(4)=0,IVCTFG(4)=0,ICLDB(4)=1            ,;EXTRA INFORMATION FLAGS
BKCNTR(1,4)=-75.                             ,;LOWER CONTOUR LIMIT
```

```
BKCNTR(2,4)=55.                                  ,;UPPER CONTOUR LIMIT
BKCNTR(3,4)=5.                                   ,;CONTOUR INTERVAL

                         ;BACKGROUND 5 INFO (MICROPHYSICS)

IBKFLG(5)=0,BKTITL(1,5)=12HMICROPHYSICS          ,;FIELD DEFINITION
MICRO(5)=1,IVCTFG(5)=0,ICLDB(5)=1                ,;EXTRA INFORMATION FLAGS
BKCNTR(1,5)=0.                                   ,;LOWER CONTOUR LIMIT
BKCNTR(2,5)=0.                                   ,;UPPER CONTOUR LIMIT
BKCNTR(3,5)=0.                                   ,;CONTOUR INTERVAL

                         ;BACKGROUND 6 INFO (LIQUID WATER)

IBKFLG(6)=18                                     ,;FIELD DEFINITION
MICRO(6)=0,IVCTFG(6)=0,ICLDB(6)=1                ,;EXTRA INFORMATION FLAGS
BKCNTR(1,6)=0.                                   ,;LOWER CONTOUR LIMIT
BKCNTR(2,6)=5.E-3                                ,;UPPER CONTOUR LIMIT
BKCNTR(3,6)=1.E-4                                ,;CONTOUR INTERVAL

                   ;BACKGROUND 7 INFO (ICE CONTENT MIX RATIO)

IBKFLG(7)=19                                     ,;FIELD DEFINITION
MICRO(7)=0,IVCTFG(7)=0,ICLDB(7)=0                ,;EXTRA INFORMATION FLAGS
BKCNTR(1,7)=0.                                   ,;LOWER CONTOUR LIMIT
BKCNTR(2,7)=5.E-3                                ,;UPPER CONTOUR LIMIT
BKCNTR(3,7)=5.E-4                                ,;CONTOUR INTERVAL

 $
;EOF
\EOD
```

APPENDIX 7 - ARCHIVED FILES ON THE CRAY ACCOUNT 1629

DEEP
LIBRARIES
GJT83
gjtlib
cgjtlib
bgjtlib
RAMS84
ramslb
ramlib
calpack
bcal
calpackSrams
fft99Srams
gjtlibSlibrary
iocnvlbSrams
misclibSrams
newlibsSrams
pltlb2Srams
pltlibSrams
ramlbSdayf
ramslbSjob
NCAR84
ncarlib3
bncarlib
ncartape9
PP
PP1GT83
pp1gt
pp1
pp1ftn
PP2GT84
pp2gt
pp2
pp2ftn
ACOUMDL983
ACOUMDL
acoustc
cnfig
cnvinitp
dryphys
initlz
metwve
model
thermo
wrapup
COMMDL
mcropkg
utility
ANLMDL
anlpk1

```
        anlpk2
        anlpk21
        anlpk3
        ncarlib$dummy
    RUNS
        CB26JUL
            HIST
            ANLS
            META
                pltfil
                pltfila
                pltfilb
                pltfilc
                ncarplta
                pltfile
            ANLJOB
                analsys3
        ATOMIC84
            HIST
            ANLS
            METAAFWL
                ncaratom
                ncaratom1
            OUTPUTS
            ANLJOB
                anlatomsym
                anlatom1
            atomicjob
        JOBMISC
            space
            untrp1
            anvil
            rauber
            rauberjobd
        TERRAIN
            terrold
        META
        ANLJOB
RAMS584
    INIT
        cnv2
        hh2
        var2
        driver2
    RAMS
        cnfig2$rams
    MODEL
        acopk2
        cupkg2
        driver2
        dryp2
        hydpk2
        micro2
        radpk2
```

- sigp2
- surf2
- surf3
- turbf2

ASSIM
- cnfig1
- stage1
- stage2
- stage3
- stage4
- stage5

RUNS
- BUMP2DS
  - BUMP2D
    - HIST
      - hbumpa
      - hbmp2a
      - hbmp2b
      - hbmp3a
      - hbmp3b
      - hbmp3c
      - hbmp3d
      - hbmp3e
    - ANLS
      - abumpa
      - abmp2a
      - abmp2b
      - abmp3a
      - abmp3b
      - abmp3c
      - abmp3d
      - abmp3e
      - abmp3f
      - abmp3g
    - bump2dlib
    - bump3lib
    - bump2djob
    - OUTPUTS
      - bmp7200
      - bmp3600
      - bmp10800
    - ANLJOB
      - b2danl
    - META
      - mtbmp3a
      - mtbmp3b
      - mtbmp3d
      - mtbmp3e
      - e
      - f
- BUMP2DN
  - HIST
    - hbmpna
    - hbmpnb

hbmpnc
ANLS
abmpna
abmpnb
abmpnc
abmpnd
abmpne
bump2dnlib
JOB
bump2dn
BUMP2DL
bump2dllib
BUMP2DL$JOB
bump2dl
B2DANL
ANLS
abmpla
abmplb
abmplc
abmpld
abmple
abmplf
abmplg
abmplh
HIST
hbmpla
hbmplb
hbmplc
hbmpld
hbmple
ANALYSIS$JOB
b2danl$job
b2danold$job
BUMPWAVE
lib$bmpwave
OUTPUTS
bwve900
bwve1800
bwve3600
bwv10800
ANLSFILES
abwvea
abwveb
abwvec
abwved
HISTFILES
hbwvea
hbwveb
hbwvec
JOBS
bmpwave
META
mtwvea
mtwveb

mtwvec
mtwved
BUMPCALM
lib$bmpcalm
JOBS
bmpcalm
OUTPUTS
bclm900
bclm7200
ANLS
abclma
abclmb
abclmc
abclmd
abc20a
abc20b
abc20c
abc20d
abc20e
abc20f
abc20g
HIST
hbclma
hbclmb
hbclmc
hbc20a
hbc20b
hbc20c
hbc20d
hbc20e%
META
mtclma
mtclmb
mtclmc
mtclmd
a20
b20
c20
d%
e20
f20
g%
BUMPWAVN
OUTPUTS
bwvn3600
bwvn7200
ANLSFILES
abwvna
abwvnb
abwvnc
abwvnd
HISTFILES
hbwvna
hbwvnb

```
    hbwvnc
  META
    mtwvna
    mtwvnb
    mtwvnc
  bumpwavnlib
  JOBS
    bmpwavn
BUMP2DH20
  ANLS
    abh20a
    abh20b
    abh20c
    abh20d
    abh20e
    abh20f
    abh20g
  HIST
  META
    a
    b
    c
    d
    e
    f
    g
  hbh20a
  hbh20b
  hbh20c
  hbh20d
  hbh20e
  JOB
    b2dh20
    b2danl
  LIB
    b2dh20
BUMP2DH25
  ANLS
    abh25a
    abh25b
    abh25c
    abh25d
    abh25e
  HIST
    hbh25a
    hbh25b
    hbh25c
    hbh25d
  META
    a
    b
    c
    d
    e
```

```
    JOB
      b2dh25
    LIB
      b2dh25
  BUMP2DW8
    HIST
      hb2w8a
      hb2w8b
      hb2w8c
      hb2w8d
    ANLS
      ab2w8a
      ab2w8b
      ab2w8c
      ab2w8d
      ab2w8e
    META
      a
      b
      c
      d
      e%
    JOB
      b2dw8
    LIB
      b2dw8
  BUMP2DW12
    META
      a
      b
      c
    ANLS
      abw12a
      abw12b
      abw12c
      abw12d
    HIST
      hbw12a
      hbw12b
      hbw12c
    abw12a
    abw12b
    abw12c
    JOB
      b2dw12
    LIB
      b2dw12
  BUMP2DI30
    JOB
    HIST
      hbi30a
      hbi30b
      hbi30c
    ANLS
```

```
      abi30a
      abi30b
      abi30c
      abi30d
    META
      a
      b
      c
      d
      e%
  BUMP2DK4I
    JOB
      ibanlk4i
    LIB
    HIST
      hbk4ia
      hbk4ib
      hbk4ic
      hbk4id
    ANLS
      abk4ia
      abk4ib
      abk4ic
      abk4id
      abk4ie
    META
      c
      d
      e%
NUKE3DBURN
  NUKE3D
    HIST
      hnukeb
      hnukec
      hnukea
      hnuked
      hnukebo
      hnukeco
      hnukeao
      hnud2s%
      hnukec2s
    ANLS
      anukea
      anukeb
      anukec
      anuked
      anukee
      anukef
      anukeao
      anukebo
      anukeco
      anukedo
      anukeco
      anukec2s
```

anuked2s
anukee2s
anuf2s%
OUTPUTS
nuk1500
META
NCAR
AFWL
mtnukeall
mtnukeb
mtnukec
nuke3dlib
nuke3djob
MISC
goodrun
oka
okb
trace1
trace2
ANLJOB
nukanl0
inukanl
nukan3lib
nukanl
nukan3
nuke3djob
NUKEDRY
HIST
hnukdb
hnukdc
hnukdd
hnukda
ANLS
anukda
anukdb
anukdc
anukdd
anukde
anukdf
OUTPUT
nukd1200
nukd1800
nukd900
JOB
ANLJOB
nukdanl
META
fire2d15y
fire2d15x
fire3d15
fire3d15r
fire3d14-15
fire3d0-3
mtnukdall

mtnukdb
mtnukdd
nukdrylib
NUKEDA
HIST
hnudaa
hnudab
hnudac
hnudad
ANLS
anudaa
anudab
anudac
anudad
anudae
anudaf
JOBS
nukeda$lib
META
mtanudaf
NUKPT1
JOB
LIB
nukpt1
HIST
hnup1a
hnup1b
hnup1c
hnup1d
hnup1e
hnup1f
hnup1g
ANLS
anup1a
anup1b
anup1c
anup1d
anup1e
anup1f
anup1g
anup1h
anup1i
anup1j
META
OUTPUTS
np2400
NUKCCN
ANLS$HIST
anuccc
anucce
anuccf
anucca
anuccd
hnucca

```
            hnuccb
            hnuccc
            hnuccd
         LIB
            nukccn
   NUKE2DBURN
      FIRES
         OUTPUTS
            fire2d
         HIST
            hfir3
            h2dspa
         ANLS
            afir3
            a2dspa
         META
         fire3lib
         JOBS
            fire
            fire3
            fire3d
            fire4
   HILL2DS
      BIGHILL
         HIST
            hhilla
            hhill1
            hhill2
            hhill3
            hhill4
            hhill5
            hhilg1
            hhilg2
            hhilg3
            hhilg4
            hhilg5
         ANLS
            ahilla
            ahill1
            ahill2
            ahill3
            ahill4
            ahill5
            ahill6
            ahill7
            ahilg1
            ahilg2
            ahilg3
            ahilg4
            ahilg5
            ahilg6
            ahilg7
         bighilllib
         bighilljob
```

```
    ANLJOB
      anl3
      anl4
    META
      mthilg1
      mthilg2
      mthilg3
      mthilg4
      mthilg7
      mthilg5
      mthilg6
  BIGHTD
    HIST
      hbhtd1
      hbhtd2
      hbhtd3
      hbhtd4
      hbhtd5
      hbhtl1
      hbhtl2
      hbhtl3
      hbhtl4
      hbhtl5
    ANLS
      abhtd1
      abhtd2
      abhtd3
      abhtd4
      abhtd5
      abhtd6
      abhtd7
      abhtl7
      abhtl1
      abhtl2
      abhtl3
      abhtl4
      abhtl5
      abhtl6
    bightdlib
    bightdjob
    META
      mtbhtl7
      mtbhtl1
      mtbhtl2
      mtbhtl3
      mtbhtl4
      mtbhtl5
      mtbhtl6
  BIGCNV
    HIST
      hbcnv1
      hbcnv2
      hbcnv3
      hbcnv4
```

hbcnv5
hbcnl1
hbcnl2
hbcnl3
hbcnl4
hbcnl5
ANLS
abcnv1
abcnv2
abcnv3
abcnv4
abcnv5
abcnv6
abcnv7
abcnl7
abcnl1
abcnl2
abcnl3
abcnl4
abcnl5
abcnl6
bigcnvjob
bigcnvlib
META
mtcnv4
mtcnv7
mtbcnl7
mtbcnl1
mtbcnl2
mtbcnl3
mtbcnl4
mtbcnl5
mtbcnl6
BIGINT
HIST
hbint1
hbint2
hbint3
ANLS
abint1
abint2
abint3
abint4
bigintlib
bigintjob
anlint
META
mtint3
BIGNF
HIST
hbnf1
hbnf2
ANLS
abnf1

```
            abnf2
          bignflib
          bignfjob
        BIGNOF
          HIST
            hbnof1
            hbnof2
          ANLS
            abnof1
            abnof2
            abnof3
          bignoflib
          META
            mtnof2
            mtnof3
          bignofjob
      CSUOROG
        HIST
          horog1
          horog2
          horog3
          horog4
        ANLS
          aorog1
          aorog2
          aorog3
          aorog4
        META
          csumetaa
          csumetab
          csumetac
          csumetad
        csu-ter
        csu-anl
        anljob
        terjob
      WAVE-CRASH
        JOBS
        LIBS
          crshanl
          wavedw
        HIST
          hcrsh1
          hcrsh2
        ANLS
          acrsh1
          acrsh2
          acrsh3
      WAVE-LID
        HIST
          hcrsh1
          hcrsh2
          hcrsh3
        ANLS
```

```
      acrsh1
      acrsh2
      acrsh3
    LIBS
      crshanl
      wavedw
    crshmta
  CCOPES
    HIST
      hcope1
      hcopa1
      hcopb1
      hcopc1
      hcopd1
      hcopf1
      hcopg1
      hcoph1
      hcopi1
      hcopj1
      hcopk1
      hcopl1
      hcopm1
      hcopn1
    ANLS
      acope1
      acopb1
      acgro1
      acopa1
      acopa2
      acopb2
      acopc1
      acopc2
      acopd1
      acopd2
      acope2
      acopf1
      acopf2
      acopg1
      acopg2
      acoph1
      acoph2
      acopi1
      acopi2
      acopj1
      acopj2
      acopk1
      acopk2
      acopl1
      acopl2
      acopm1
      acopm2
      acopn1
      acopn2
  JOBS
```

- ccope
- ccopejob
- ccopexjob
- copanljob
- cjanljob
- ckanljob
- ccopexlib
- copanllib
- cjanllib
- ckanllib

- META
  - mtcop1
  - mtcopb1
  - mtgro1
  - mtcopn1
  - mtcopm1
  - mtcopl1
  - mtcopa2
  - mtcopb2
  - mtcopc2
  - mtcopd2
  - mtcope2
  - mtcopf2
  - mtcopg2
  - mtcoph2
  - mtcopc1
  - mtcopd1
  - mtcopa1a
  - mtcopa1b
  - mtcope1
  - mtcopf1
  - mtcopg1
  - mtcopi1
  - mtcopj1
  - mtcopk1
  - mtcopi2
  - mtcopj2
  - mtcopl2
  - mtcopm2
  - mtcopn2
  - mtcopk2
  - mtcopc1c
  - mtcopc1d
  - mtcoph1c

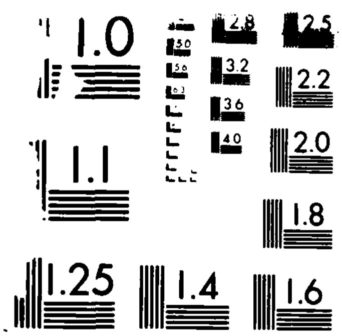
CSUOROG1

- HIST
  - horo11
  - horo12
  - horo13
  - horo14
  - horo15
  - horo25
  - horo26
- ANLS

aoro11
aoro12
aoro13
aror14
aoro15
aoro25
aoro26
META
csumta
csumta11
csumta12
csumta13
csumta14
csumta15
csumta16
csumta16a
csumta24
csumta25
csumta26
ANLMDL
cnfigold
anlpk1a
anlpk2a
anlpk3a
sound
uwcalc
cnfiga
IBMSTUFF
EXEC
b
bdmp1
c1
findfile
flop
mod1
outgau
cracc
execute
mdlprt
mctprint
catlog
catprt
cf80
clear
clrscrn
crayrun
dateconv
linkto
mod
octal
profile
profile1
rdr
seq

AD-A175 489 CLOUD MICROPHYSICS ANALYSIS AND MODELING(U) OPHIR CORP 2/2
LAKEWOOD CO L D NELSON ET AL. JUL 86 AFGL-TR-86-0179
F19628-83-C-0130

UNCLASSIFIED F/G 4/2 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

```
    usernum
    linum
    m
    c
    ncar2gli
  XEDIT
    cracc
    execute
    mod1
    mod2
    mod3
    mod4
    mod5
    strtst
    subsort
    mod6
    aprf
    aprof
    jclmcro
    profile
    scanfile
  MISC
    syns$synonym
    sysu$txtlib
    ncar2$script
    ncar2$txtlib
MISC
  MORTS82
    profiledata
    profile
    warmcu
  rundocs
  opdoc
NETED
  neted
RAMS586
  RUNS
    HILL2DS
      BIG5B
        big5bjob
        ANLS
        HIST
        META
        5banljob
    CCOPE2DS
      CC5
        cc5job
        HIST
        ANLS
        META
        cc5anljob
  INIT
    conv5
    driver5
```

hh5
RAMS
cnfig5
misclib
MODEL
acopk5
cycl5
dryp5
hydpk5
micro5
radpk5
sigp5
driver5
surf5
turbf5
ANLMDL
anlpka2
anlpka3
anlpkb2
anlpkc2
anlpkd2
anlpke2
cnfig2
cnfig3

APPENDIX 8 - THE CONTENTS OF THE TAPE SUE023

---START OF ENCLOSURE---

DATE:12/09/83 TIME:11:31:40 SEQUENCE:SB5975

NCAR SOFTWARE DISTRIBUTION PACKAGE FOR NELSON

MATERIALS SENT: TAPE

TAPE NAME: SUE023

TAPE FORMAT: 9 TRACK, 1600 BPI, ASCII, ODD PARITY, UNLABELLED

Each physical record is 24 card images (1920 characters) except for the last record of each file, which may be shorter.

Files are separated by file marks.

The files are repeated after the last file, in case any of them are unreadable. Two file marks follow the last repeated file.

The first card of each file is a comment card with the name, source, and version, except for data files, which will not have such a comment card.

Every card of each file (except for data files) contains a sequence number in columns 72 through 80.

File name and source of library files are provided in the table below to uniquely identify file versions in case of any future inquiry to NCAR about the files sent in this distribution package.

A source of testlib indicates the file is a portable test driver that can be called to verify that the corresponding library file has been correctly implemented.

FILE	FILE NAME	SOURCE	CARDS,	NUMBER OF
1	CONRCQCK	CRAYLIB	942	HEADERSEQUENCE
2	CDNREC	CRAYLIB	1108	HEADERSEQUENCE
3	DASHSUPR	CRAYLIB	2642	HEADERSEQUENCE
4	EZMAP	CRAYLIB	2831	HEADERSEQUENCE
5	MACHCR	CRAYLIB	198	HEADERSEQUENCE
6	MCTRPRNT	CRAYLIB	1932	HEADERSEQUENCE
7	PLOT88	CRAYLIB	3206	HEADERSEQUENCE
8	PWRITX	CRAYLIB	2722	HEADERSEQUENCE
9	PWRITXNT	CRAYLIB	1239	HEADERSEQUENCE
10	SPPRT12C	CRAYLIB	158	HEADERSEQUENCE
11	SPPRT12F	CRAYLIB	103	HEADERSEQUENCE
12	ENCD	PORTLIB	76	HEADERSEQUENCE

13	ERPORT	PORTLIB	308	HEADERSEQUENCE
14	I1MACH	PORTLIB	360	HEADERSEQUENCE
15	MCTRPORT	PORTLIB	1802	HEADERSEQUENCE
16	MCTRPRNP	PORTLIB	1928	HEADERSEQUENCE
17	PWRITXC1	PORTLIB	49	
18	PWRITXC2	PORTLIB	575	
19	PWRITXD1	PORTLIB	49	
20	PWRITXD2	PORTLIB	575	
21	Q8QST4	PORTLIB	25	HEADERSEQUENCE
22	R1MACH	PORTLIB	161	HEADERSEQUENCE
23	SUPMAP	PORTLIB	1813	HEADERSEQUENCE
24	SUPMAPDT	PORTLIB	10843	
25	TEST12	PORTLIB	2446	HEADERSEQUENCE
26	TESTPLOT	PORTLIB	209	HEADERSEQUENCE
27	TESTSPP	PORTLIB	439	HEADERSEQUENCE
28	TRUNC	PORTLIB	22	HEADERSEQUENCE
29	ULIBER	PORTLIB	54	HEADERSEQUENCE
30	AUTOGRPH	TESTLIB	188	HEADERSEQUENCE
31	CNRCSMTH	TESTLIB	117	HEADERSEQUENCE
32	CONRAN	TESTLIB	156	HEADERSEQUENCE
33	CONRAQ	TESTLIB	147	HEADERSEQUENCE
34	CONRAS	TESTLIB	149	HEADERSEQUENCE
35	CONRCQCK	TESTLIB	117	HEADERSEQUENCE
36	CONRCSPR	TESTLIB	117	HEADERSEQUENCE
37	CONREC	TESTLIB	117	HEADERSEQUENCE
38	DASHCHAR	TESTLIB	147	HEADERSEQUENCE
39	DASHLINE	TESTLIB	142	HEADER SEQUENCE
40	DASHSMTH	TESTLIB	147	HEADERSEQUENCE
41	DASHSUPR	TESTLIB	154	HEADERSEQUENCE
42	HAFTON	TESTLIB	128	HEADERSEQUENCE
43	ISOSRF	TESTLIB	130	HEADERSEQUENCE
44	ISISRFHR	TESTLIB	167	HEADERSEQUENCE
45	PWRITX	TESTLIB	163	HEADERSEQUENCE
46	PWRY	TESTLIB	90	HEADERSEQUENCE
47	PWRZI	TESTLIB	129	HEADERSEQUENCE
48	PWRZS	TESTLIB	123	HEADERSEQUENCE
49	PWRZT	TESTLIB	115	HEADERSEQUENCE
50	SCROLL	TESTLIB	97	HEADERSEQUENCE
51	SRFACE	TESTLIB	138	HEADERSEQUENCE
52	STRMLN	TESTLIB	92	HEADERSEQUENCE
53	SUPMAP	TESTLIB	277	HEADERSEQUENCE
54	THREED	TESTLIB	127	HEADERSEQUENCE
55	VELVCT	TESTLIB	117	HEADERSEQUENCE
56	WINDOW	TESTLIB	129	HEADERSEQUENCE
57	AUTOGRPH	ULIB	6667	HEADERSEQUENCE
58	CONCOM	ULIB	1753	HEADERSEQUENCE
59	CONRAN	ULIB	1778	HEADERSEQUENCE
60	CONRAQ	ULIB	1502	HEADERSEQUENCE
61	CONRAS	ULIB	1792	HEADERSEQUENCE
62	CONRCSPR	ULIB	3711	HEADERSEQUENCE
63	CONTERP	ULIB	3351	HEADERSEQUENCE
64	DASHCHAR	ULIB	1123	HEADERSEQUENCE
65	DASHLINE	ULIB	548	HEADERSEQUENCE
66	DASHSMTH	ULIB	1903	HEADERSEQUENCE

67	HAFTON	ULIB	777	HEADERSEQUENCE
68	ISOSRF	ULIB	2197	HEADERSEQUENCE
69	ISOSRFHR	ULIB	577	HEADERSEQUENCE
70	PWRY	ULIB	677	HEADERSEQUENCE
71	PWRZI	ULIB	708	HEADERSEQUENCE
72	PWRZS	ULIB	711	HEADERSEQUENCE
73	PWRZT	ULIB	708	HEADERSEQUENCE
74	SCROLL	ULIB	2092	HEADERSEQUENCE
75	SRFACE	ULIB	1266	HEADERSEQUENCE
76	STRMLN	ULIB	959	HEADERSEQUENCE
77	THREED	ULIB	660	HEADERSEQUENCE
78	VELVCT	ULIB	749	HEADERSEQUENCE
79	WINDOW	ULIB	1134	HEADERSEQUENCE

---END OF ENCLOSURE---

APPENDIX 9 - DATA STATEMENTS CHANGED IN TEST12

WRDLNG contains the number of bits in a machine word

```
DATA WRDLNG/64/
```

NCHAR is the number of characters in a machine word

```
DATA NCHAR/8/
```

INLNGA is the number of bits in an integer assignment

```
DATA INLNGA/63/
```

INLNGC is the number of bits in an integer comparison

```
DATA INLNGC/63/
```

LIMPOS is a large positive integer

```
DATA LIMPOS/9223372036854775807/
```

LIMNEG is a very negative integer (in magnitude)

```
DATA LIMNEG/1000000000000000000000B/
```

BIGRL is a large positive real number

```
DATA BIGRL/1.E2000/
```

SMLRL is small positive real number

```
DATA SMLRL/1.E-2000/
```

The MASK0 and MASK1 data cards were commented out and their values are generated using:

```
      DO 1000 I=1,63
      MASK0(I)=2**(I-1)
      WRITE(6,1001)MASK0(I)
1001  FORMAT(I23)
1000  CONTINUE
      MASK0(64)=1000000000000000000000B
      WRITE(6,1003)MASK0(64)
1003  FORMAT(I23)
```

and

```
      DO 1005 I=1,64
      MASK1(I)=(2**(I-1))-1
      WRITE(6,1006)MASK1(I)
1006  FORMAT(I23)
100   CONTINUE
```

```
      MASK1(65)=1111111111111111111111B
      WRITE(6,1008)MASK1(65)
1008  FORMAT(I23)
```

APPENDIX 10 - CHANGES MADE TO TRNSLATE

A program card was added to the beginning of the file

```
PROGRAM MAIN(TAPE98,TAPE99,TAPE6)
```

The following data statements were changed:

The number of bits in a default length integer variable

```
DATA NBPW/64/
```

The number of the unit containing metacode

```
DATA NUMC/98/
```

The flag to indicate character code used in metacode

```
DATA NCOD/1/
```

The type of numbers passed to the plotter interface

```
DATA NTYP/1/
```

The minimum x addressed to be produced

```
DATA XMIN/0./
```

The maximum x addresses to be produced

```
DATA XMAX/1./
```

The minimum y addresses to be produced

```
DATA YMIN/0./
```

The maximum y addresses to be produced

```
DATA YMAX/1./
```

The orientation of the picture

```
DATA IOREN/0/
```

The commands

```
CALL PLOTS(100.,0.,99)
CALL SCREEN(0.,1.,0.,1.)
CALL VWPORT(0.,1.,0.,1.)
CALL WINDOW(0.,1.,0.,1.)
```

were added after the comment

```
C       CHECK FOR PROPER IMPLEMENTATION
```

The command

```
CALL PLOT(0,0,40)
```

was added after the statement

```
WRITE(NWRT,1001)NPIC
```

And at the end of SUBROUTINE PLOTMC, add

```
IF(NPEN.EQ.0)K=3
IF(NPEN.EQ.1)K=2
IF(NPEN.EQ.2)K=10
IF(NPEN.EQ.-1)K=-3
CALL PLOT(X,Y,K)
```

APPENDIX 11 - CREATING THE BINARY FILE TAPE9

To create the binary file TAPE9 which was needed to run the routines given in the file PWRTX of CRAYLIB, five other files from the NCAR Graphics System tape were needed. They were PWRITXNT of CRAYLIB and PWRITXC1, PWRITXC2, PWRITXD1, and PWRITXD2 all of PORTLIB. PWRITXC1, PWRITXC2, PWRITXD1, and PWRITXD2 were given the local filenames on the AFWL Cray of TAPE1, TAPE2, TAPE3, and TAPE4 respectively. The file PWRITXNT contained the program which creates TAPE9, using TAPE1, TAPE2, TAPE3, and TAPE4. A program card was addded to PWRITXNT

```
PROGRAM PCRBIN(OUTPUT,TAPE6=OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE9)
```

PWRITXNT was combined with the 14 basic Fortran routines in the files TESTRY and ROUTINES

```
COMBINE PTEST PWRITXNT TEXTRY ROUTINES <cr>
```

It was then compiled, loaded, and run to create the local binary file TAPE9.

```
CFT I=PTEST,B=BTEST <cr>
LDR B=BTEST <cr>
XBTEST <cr>
```

APPENDIX 12 - PROGRAM TESTLIB

```
      PROGRAM TESTLIB(INPUT,OUTPUT,TAPE6=OUTPUT,TAPE98,TAPE9)
C     THIS PROGRAM TESTS THE ROUTINES CNRCSMTH, CONREC,
C     DASHCHAR, HAFTON, AND PWRITX.
      CALL TCNSMT(0)
      CALL TCONRE(0)
      CALL TDASHC(0)
      CALL THAFTO(0)
      CALL TPWRTX(0)
      STOP
      END
```

APPENDIX 13 - SAMPLE SESSION WITH AUTOSUM

```
MASS GET NATIVE:/AUTOLOG/ODD/V
4/01/26 12:45:37.773  get native:/autolog/odd/v
001 (53133400b bits) 84/01/26 00:23:47.421

 all done
AUTOSUM
><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*>
 type first selection criterion:
? CHARGE-CODE=0000XXXX
 additional criterion - cr if none.
? SYSTEM=CTSS
 additional criterion - cr if none.
?
 up to three break criteria:
? DATE SHIFT USER-NUMBER
ave selected entries on a file type 'yes'.
? YES
eria for display?  (up to five words or cr):
? SERVICE-UNITS CP-HRS DOLLARS DURATION PRIORITY

 >>>>> autosum version 6 jan 83- run on machine v  01/26/84  12:45:17

 >>>>> run identification:
? PM 26 JAN 84

 >>>>> selection criteria:
      charge  =  XXXXXXXX
      system  =  ctss

 >>>>> break criteria:  date     shift    user

 >>>>> selected entries written on file 'select' (or its replacement).

 >>>>> selected    56 of 10968 total entries.

 type 'yes' to suppress listing at terminal - cr otherwise.
?
```

source of data: ctss

s-units	cp-hrs	dollars	duration	priority	date	shift	user
0.01	0.00	23.04	2.21	2.00	84/01/03	d	001630
0.00	0.00	0.54	2.11	1.00	84/01/03	d	001631
0.01	0.00	9.90	1.13	2.00	84/01/03	n	001630
0.02	0.04	40.68	0.87	2.00	84/01/03	n	001631
0.01	0.00	16.20	1.27	2.00	84/01/04	d	01630

0.00	0.00	7.20	0.61	2.00	84/01/04	n	001630
0.02	0.01	36.18	3.42	2.00	84/01/05	d	001630
0.13	0.12	238.32	6.23	2.00	84/01/05	d	001631
0.05	0.03	93.78	3.84	2.00	84/01/06	d	001630
0.00	0.00	8.46	1.50	1.00	84/01/06	d	001631
0.00	0.00	0.18	0.67	1.00	84/01/06	n	001630
0.02	0.00	30.78	2.67	1.00	84/01/09	d	001630
0.00	0.00	0.00	0.50	1.00	84/01/09	w	001630
0.00	0.00	0.72	0.99	1.00	84/01/12	d	001631
0.00	0.01	3.60	0.11	1.00	84/01/14	w	001630
0.00	0.00	6.66	1.00	2.00	84/01/16	d	001631
0.01	0.01	14.22	1.00	1.00	84/01/16	n	001631
0.00	0.00	0.36	0.50	1.00	84/01/16	w	001630
0.02	0.01	34.02	1.48	1.00	84/01/17	d	001630
0.05	0.04	84.06	2.00	1.00	84/01/17	d	001631
0.02	0.01	40.68	2.50	1.00	84/01/18	d	001629
0.02	0.01	33.30	1.50	2.00	84/01/18	d	001630
0.03	0.03	55.98	1.00	1.00	84/01/18	d	001631
0.00	0.00	1.80	1.00	1.00	84/01/18	n	001630
0.11	0.12	196.02	3.75	1.00	84/01/19	d	001629
0.00	0.00	9.00	1.50	1.00	84/01/19	d	001631
0.00	0.00	0.36	1.05	1.00	84/01/20	d	001629
0.02	0.01	32.40	2.04	2.00	84/01/20	d	001630
0.01	0.01	17.10	0.93	2.00	84/01/20	n	001630
0.06	0.11	104.58	1.23	2.00	84/01/20	n	001631
0.18	0.11	322.02	3.68	2.00	84/01/23	d	001630
0.01	0.01	10.98	1.50	2.00	84/01/23	d	001631

0.02	0.03	43.74	0.67	1.00	84/01/23	n 001630
0.00	0.02	7.20	0.32	1.00	84/01/23	w 001630
0.00	0.00	1.80	0.50	1.00	84/01/24	d 001629
0.00	0.00	6.84	1.00	1.00	84/01/24	d 001631
0.02	0.04	43.20	0.97	1.00	84/01/24	n 001630
0.10	0.06	182.34	5.56	2.00	84/01/25	d 001630
0.00	0.00	7.02	0.73	1.00	84/01/25	d 001631
0.00	0.01	9.00	0.10	1.00	84/01/25	n 001630
0.99	0.90	1774.26	65.65	56.00	totals	

```
Rates are printed in the output file.
end autosum.

type 'end' or 'more'.
? END
  777
autosum  ctss time   8.209   seconds
cpu=  5.634  sys=   .051  i/o+memory=   2.524

all done
```

APPENDIX 14 - T1 DATA OUTLINE FOR NEW TAPE

Format of the University of North Dakota Cessna Citation II Aircraft Data Tapes.

These nine-track magnetic tapes recorded at 1600 BPI contain two sets of weather information. They are called T1 and T24 data. T1 is data collected at the rate of 0.98304 seconds or is an average of T24 data samples. T24 is data collected at the rate of 0.04096 seconds.

T1 Header Record

NAME	VALUES	TYPE	BYTES	MIN.	MAX.
DAY	1	INTEGER	4	1	31
MONTH	1	INTEGER	4	1	12
YEAR	1	INTEGER	4	0	99
RECORD TYPE	1	INTEGER	4	1	1
RECORD LENGTH	1	INTEGER	4	@17940	@17940
VERSION NO.	1	INTEGER	4	0	1
CHANNEL CONSTANTS	64	REALS	4	4-	-
DTAS CONSTANTS	2	REALS	4	-	-
PITOT FLAGS	1	INTEGER	4	0	1
SPARE	27	INTEGER	4	-	-

Subsequent records on the T1 file are arranged into groups of three records. There are sixty samples of each value in a group. The data rate is 0.98304 seconds or the average of T24 data.

A GROUP: 3 T1 DATA RECORDS

NOTE: The record length maximum is 7200 and the minimum is 400 bytes. This is because the original data has been rewritten and reduced inorder to read on the AFGL VAX.

RECORD 1 (ALL ARE 4 BYTE REALS)

NAME	UNITS	VALUES	MIN.	MAX.
BINARY DRIFT	-	60	-	-
DRIFT ANGLE	DEGREES	60	-39.9	39.9
ANALOG SPARE 1	-	60	-	-
LATITUDE MINUTES	MINUTES	60	0.0	59.9
LONGITUDE MINUTES	MINUTES	60	0.0	59.9
ANALOG SPARE 2	-	60	-	-
ANALOG SPARE 3	-	60	-	-
WIND DIRECTION	DEGREES	60	0.0	359.9
WIND VELOCITY	KNOTS	60	0.0	359.9
SPARE 3	-	120	-	-
CROSS TRACK DISTANCE	NAUTICAL MILES	60	-799.0	-799.9

RECORD 2 (ALL ARE 4 BYTE REALS)

NAME	UNITS	VALUES	MIN.	MAX.

GROUND SPEED	KNOTS	60	0.0	799.9
TRUE HEADING	DEGREES	60	0.0	359.9
TRACK ANGLE	DEGREES	60	0.0	359.9
*DRIFT ANGLE	DEGREES	60	-39.9	39.9
*TRUE HEADING	DEGREES	60	0.0	360.0
*INS HEADING	DEGREES	60	0.0	360.0
*LATITUDE MINUTES	MINUTES	60	0.0	60.0
*LONGITUDE MINUTES	MINUTES	60	0.0	60.0
*MAGNETIC HEADING	DEGREES	60	0.0	360.0
*VERTICAL ACC. GAINED	VOLTS	60	-	-
*VOR	DEGREES	60	0.0	360.0
*ATTACK ANGLE	VOLTS	60	-2.5	2.5
*SIDESLIP ANGLE	VOLTS	60	-2.5	2.5
*PITOT STATIC NOSE	VOLTS	60	0.0	10.0
*PITOT STATIC WING	VOLTS	60	0.0	10.0
*ICE RATE METER	VOLTS	60	0.0	5.0
*STATIC PRESSURE	VOLTS	60	0.0	5.0
*ROSEMOUNT	VOLTS	60	0.0	5.0
*DEWPOINT	VOLTS	60	-5.0	5.0
*REVERSE FLOW	VOLTS	60	0.0	5.0
*JW LIQUID WATER	VOLTS	60	0.0	10.0
*DME	VOLTS	60	0.0	10.0
*VERTICAL ACC.	VOLTS	60	-10.0	10.0
*PITCH COARSE	DEGREES	60	-180.0	180.0
*PITCH FINE	DEGREES	60	-45.0	45.0
*ROLL COARSE	DEGREES	60	-180.0	180.0
*ROLL FINE	DEGREES	60	-45.0	45.0
*ALTITUDE	FEET	60	0.0	79999.0
*SEC. FROM MIDNIT	SECONDS	60	0.0	86400.0
SEC. REMAINING	SECONDS	60	0.0	59.999

RECORD 3 (4 BYTE REALS)

NAME	UNITS	VALUES	MIN.	MAX.
INS TIME	SECONDS	60	0.0	79999.0
SEC. FROM MIDNIT	SECONDS	60	0.0	86399.0

(2 BYTE REALS)

NAME	UNITS	VALUES	MIN.	MAX.
SPARE 5	-	1200	-	-
LATITUDE DEG.	DEGREES	60	-90.0	90.0
LONGITUDE DEG.	DEGREES	60	-180.0	180.0
*LATITUDE DEG.	DEGREES	60	-90.0	90.0
*LONGITUDE DEG.	DEGREES	60	-180.0	180.0

(ONE BYTE EACH)

NAME	UNITS	VALUES	MIN.	MAX.
INI FLAG	-	60	-	-
INS STATUS	-	60	-	-

SPARE 6	-	60	-	-

Notes: Several recorded values need to be calibrated using the CHANNEL CONSTANTS from the header record. If we assume the 64 CHANEL CONSTANTS are in array called CALIB then the following equations convert volts to meaningful units:

(VERTICAL ACCELERATION GAINED) X CALIB(63) + CALIB(64)	produces	volts
(ATTACK ANGLE) X CALIB(1) + CALIB(2)	"	millibars
(SIDESLIP ANGLE) X CALIB(3) + CALIB(4)	"	millibars
(PITOT STATIC NOSE) X CALIB(5) + CALIB(6)	"	millibars
(PITOT STATIC WING) X CALIB(7) + CALIB(8)	"	millibars
(ICE RATE METER) X CALIB(33) + CALIB(34)	"	volts
(STATIC PRESSURE) X CALIB(11) + CALIB(12)	"	millibars
(ROSEMOUNT) X CALIB(13) + CALIB(14)	"	degrees C
(DEWPOINT) X CALIB(35) + CALIB(36)	"	degrees C
(REVERSE FLOW) X CALIB(9) + CALIB(10)	"	degrees C
(JW LIQUID WATER) X CALIB(37) + CALIB(38)	"	conc. @ 100 knots
(DME) X CALIB(39) + CALIB(40)	"	nautical miles
(VERTICAL ACCELERATION) X CALIB(41) + CALIB(42)	"	meters/sec^2

****

TABULATED USING THE AVERAGE OF 24 T24 DATA VALUES.

```
SOPTS

OPROT     = T,

OPLCD     = F,

OPPRE     = T,

OPPRT     = F,

STIME     = -.1E+01,

NEOF      = 4,

LMAX      = 160,

DEBUG     = F,

NGAP      = 1,

LMIN      = 3,

REJH      = T,

NVREJ     = 1,

TMAX      = .3E+02,

SEND

STYPDAT

PROB1     = 0.0, 0.0, 0.0, 0.0, 0.0,

VERA41    = 12,

LR1       = 4,

SEND
FLT E78-11 24MAR-7BKNE230
194300 200700
```

RECORD 9 ON FILE 2

TIME	TAS(KTS)	PRES(MB)	HED(DEG)	JW(G/M**3)	FROST(C)	TRUE(C)	IAS(KTS)	CAS(KTS)	ALT(M)	CLK PCT
194300	217.50	391.52	-108.54	.02	.14	-34.05	150.88	221.02	7333.48	99.43
194301	217.90	391.42	-109.08	.02	.14	-34.05	151.17	221.46	7335.32	
194302	218.30	391.52	-109.69	.02	.14	-34.04	151.28	221.60	7333.48	99.44
194303	218.45	391.62	-110.34	.02	.14	-34.03	151.51	221.90	7331.64	
194304	218.55	391.52	-110.95	.02	.14	-34.03	151.68	222.17	7333.48	99.44
194305	219.00	391.62	-111.56	.02	.14	-34.01	151.97	222.55	7331.64	
194306	219.40	391.62	-112.18	.02	.14	-34.01	152.14	222.80	7331.64	99.43
194307	219.40	391.62	-112.82	.02	.14	-33.98	152.08	222.73	7331.64	
194308	219.55	391.62	-113.54	.02	.14	-34.00	152.25	222.97	7331.64	99.43
194309	219.80	391.62	-114.16	.02	.14	-34.02	152.31	223.04	7331.64	

RECORD 20 ON FILE 2

TIME	TAS(KTS)	PRES(MB)	HED(DEG)	JW(G/M**3)	FROST(C)	TRUE(C)	IAS(KTS)	CAS(KTS)	ALT(M)	CLK PCT
194310	220.00	391.52	-114.84	.02	.14	-34.05	152.36	223.13	7333.48	99.42
194311	220.00	391.52	-115.60	.02	.14	-34.04	152.25	222.98	7333.48	
194312	219.95	391.52	-116.32	.02	.14	-34.06	152.25	222.96	7333.48	99.44
194313	219.85	391.42	-116.96	.02	.14	-34.09	152.14	222.82	7335.32	
194314	219.70	391.42	-117.25	.02	.14	-34.12	152.19	222.88	7335.32	99.43
194315	219.80	391.42	-117.43	.02	.14	-34.13	152.19	222.88	7335.32	
194316	219.80	391.32	-117.61	.02	.14	-34.13	152.14	222.83	7337.17	99.42
194317	219.75	391.21	-117.86	.02	.14	-34.11	152.08	222.79	7339.01	
194318	219.75	391.32	-117.94	.02	.14	-34.11	152.08	222.76	7337.17	99.43
194319	219.65	391.11	-117.94	.02	.14	-34.11	152.14	222.89	7340.86	

RECORD 31 ON FILE 2

TIME	TAS(KTS)	PRES(MB)	HED(DEG)	JW(G/M**3)	FROST(C)	TRUE(C)	IAS(KTS)	CAS(KTS)	ALT(M)	CLK PCT
194320	219.75	391.11	-117.97	.02	.14	-34.11	152.14	222.89	7340.86	99.43
194321	219.85	391.01	-117.94	.02	.14	-34.12	152.08	222.84	7342.70	
194322	219.80	391.01	-117.76	.02	.14	-34.13	152.08	222.83	7342.70	99.42
194323	219.75	391.01	-117.61	.03	.14	-34.12	152.14	222.92	7342.70	
194324	219.90	391.01	-117.58	.03	.14	-34.11	152.25	223.08	7342.70	99.43
194325	220.00	391.01	-117.61	.02	.15	-34.12	152.42	223.32	7342.70	
194326	220.10	391.11	-117.58	.02	.15	-34.15	152.65	223.60	7340.86	99.43
194327	220.50	391.01	-117.61	.02	.14	-34.17	152.82	223.86	7342.70	
194328	220.55	391.01	-117.58	.02	.14	-34.19	152.93	224.02	7342.70	99.42
194329	220.70	391.01	-117.61	.02	.14	-34.17	153.10	224.26	7342.70	

RECORD 42 ON FILE 2

TIME	TAS(KTS)	PRES(MB)	HED(DEG)	JW(G/M**3)	FROST(C)	TRUE(C)	IAS(KTS)	CAS(KTS)	ALT(M)	CLK PCT
194330	220.80	391.11	-117.58	.02	.14	-34.17	153.16	224.32	7340.86	99.43
194331	221.00	391.01	-117.61	.02	.14	-34.15	153.16	224.35	7342.70	
194332	221.25	391.11	-117.58	.02	.14	-34.16	153.32	224.56	7340.86	99.45
194333	221.35	391.01	-117.61	.02	.15	-34.16	153.32	224.59	7342.70	
194334	221.50	391.11	-117.58	.02	.14	-34.17	153.38	224.64	7340.86	99.44
194335	221.55	391.11	-117.61	.02	.14	-34.18	153.38	224.63	7340.86	
194336	221.45	391.01	-117.58	.02	.14	-34.20	153.49	224.81	7342.70	99.45
194337	221.60	391.11	-117.61	.02	.14	-34.19	153.49	224.79	7340.88	
194338	221.65	391.01	-117.58	.02	.14	-34.19	153.66	225.06	7342.70	99.46
194339	221.80	391.11	-117.61	.02	.14	-34.18	153.77	225.19	7340.86	

RECORD 53 ON FILE 2

TIME	TAS(KTS)	PRES(MB)	HED(DEG)	JW(G/M**3)	FROST(C)	TRUE(C)	IAS(KTS)	CAS(KTS)	ALT(M)	CLK PCT
194340	222.10	391.11	-117.61	.02	.15	-34.20	153.94	225.43	7340.86	99.47
194341	222.15	391.11	-117.58	.02	.14	-34.20	154.00	225.50	7340.86	
194342	222.30	391.21	-117.61	.02	.14	-34.22	154.22	225.79	7339.01	99.48
194343	222.60	391.21	-117.59	.02	.14	-34.25	154.67	226.41	7339.01	
194344	223.20	391.32	-117.43	.02	.14	-34.26	154.73	226.46	7337.17	99.45
194345	223.25	391.32	-117.25	.02	.15	-34.25	154.78	226.54	7337.17	
194346	223.25	391.42	-117.25	.02	.14	-34.25	154.84	226.59	7335.32	99.45
194347	223.30	391.42	-117.25	.02	.14	-34.27	155.06	226.90	7335.32	
194348	223.50	391.52	-117.25	.02	.14	-34.26	155.28	227.19	7333.48	99.44
194349	223.60	391.52	-117.25	.02	.15	-34.24	155.34	227.28	7333.48	

RECORD 64 ON FILE 2

TIME	TAS(KTS)	PRES(MB)	HED(DEG)	JW(G/M**3)	FROST(C)	TRUE(C)	IAS(KTS)	CAS(KTS)	ALT(M)	CLK PCT
194350	223.75	391.62	-117.25	.02	.15	-34.25	155.56	227.56	7331.64	99.46
194351	224.25	391.62	-117.25	.02	.14	-34.22	155.67	227.73	7331.64	
194352	224.45	391.73	-117.29	.02	.14	-34.22	155.84	227.94	7329.80	99.50
194353	224.65	391.73	-117.25	.02	.14	-34.21	156.01	228.19	7329.80	
194354	224.85	391.73	-117.25	.02	.15	-34.21	156.12	228.35	7329.80	99.51
194355	225.00	391.83	-117.25	.02	.14	-34.20	156.12	228.32	7327.95	
194356	225.00	391.73	-117.25	.02	.14	-34.22	156.39	228.73	7329.80	99.49
194357	225.25	391.73	-117.25	.02	.15	-34.23	156.56	228.96	7329.80	
194358	225.40	391.93	-117.14	.02	.14	-34.25	156.67	229.05	7326.11	99.49
194359	225.60	391.83	-116.93	.02	.14	-34.25	156.61	229.01	7327.95	

RECORD 75 ON FILE 2

TIME	TAS(KTS)	PRES(MB)	HED(DEG)	JW(G/M**3)	FROST(C)	TRUE(C)	IAS(KTS)	CAS(KTS)	ALT(M)	CLK PCT
194400	225.35	391.83	-116.89	.02	.15	-34.25	156.56	228.92	7327.95	99.50
194401	225.55	391.93	-116.89	.02	.14	-34.26	156.50	228.81	7326.11	
194402	225.35	391.83	-116.89	.02	.14	-34.27	156.61	228.99	7327.95	99.49
194403	225.50	391.83	-117.40	.02	.15	-34.27	156.67	229.07	7327.95	
194404	225.05	391.83	-118.76	.02	.14	-34.25	156.50	228.85	7327.95	99.49
194405	225.55	391.83	-120.60	.02	.15	-34.25	156.50	228.85	7327.95	
194406	225.55	391.93	-122.15	.02	.15	-34.23	156.45	228.75	7326.11	99.48
194407	225.50	391.83	-123.16	.02	.14	-34.21	156.50	228.87	7327.95	
194408	225.40	391.93	-123.84	.02	.15	-34.18	156.50	228.85	7326.11	99.49
194409	225.45	391.93	-123.91	.02	.14	-34.19	156.61	229.00	7326.11	

RECORD 86 ON FILE 2

TIME	TAS(KTS)	PRES(MB)	HED(DEG)	JW(G/M**3)	FROST(C)	TRUE(C)	IAS(KTS)	CAS(KTS)	ALT(M)	CLK PCT
194410	225.50	392.03	-123.84	.02	.14	-34.19	156.78	229.21	7324.27	99.50
194411	225.75	391.93	-123.08	.02	.15	-34.18	156.83	229.32	7326.11	
194412	225.95	392.03	-122.54	.02	.14	-34.15	156.83	229.31	7324.27	99.49
194413	226.00	392.03	-122.54	.02	.14	-34.12	156.83	229.32	7324.27	
194414	226.05	392.03	-122.54	.02	.15	-34.12	156.89	229.40	7324.27	99.50
194415	226.25	392.03	-122.72	.03	.14	-34.12	156.83	229.32	7324.27	
194416	225.90	391.93	-122.87	.02	.15	-34.14	156.72	229.18	7326.11	99.48
194417	225.60	391.93	-122.90	.02	.14	-34.18	156.89	229.40	7326.11	
194418	225.95	392.03	-122.87	.02	.15	-34.19	156.89	229.37	7324.27	99.49
194419	226.00	392.03	-122.83	.02	.14	-34.18	156.94	229.45	7324.27	

APPENDIX 16 - INPUT VARIABLES AND SAMPLE OUTPUT (OPTION 5)

(this page intentionally left blank)

```
$OPTS
OPROT    = T,
OPCLD    = F,
OPPRE    = T,
OPPRT    = F,
STIME    = -.1E+01,
NEOF     = 4,
LMAX     = 160,
DEBUG    = F,
NGAP     = 1,
LMIN     = 3,
REJH     = T,
NVREJ    = 1,
TMAX     = .3E+02,
$END

$TYPDAT
PROB1    = 0.0, 0.0, 0.0, 0.0, 0.0,
VERA41   = 12,
LR1      = 5,
$END
FLT E78-11 24MAR-78KNE230
194300 200700
```

```
FLT E78-... ON 24MAR-78K              194300 TO 200700                PAGE  2

                                  PASS TOTALS
DENDRITES      1  NEEDLES     518  COLUMNS      94  PLATES      329  STREAKERS      0  MISC.

  5388 DATA HISTORIES WERE REJECTED BECAUSE THEY TOUCHED THE SIDE
    54 DATA HISTORIES WERE REJECTED BECAUSE THEY WERE INCOMPLETE
     0 DATA HISTORIES WERE REJECTED BECAUSE THEY WERE LONGER THAN  160 DIODES
     0 DATA    RECORDS WERE REJECTED BECAUSE THE TEMPERATURE WAS GREATER THAN  30.0 DEGREES
107929 DATA HISTORIES WERE REJECTED BECAUSE THE PARTICLE WAS  LESS THAN     3 DIODES
  1608 DATA HISTORIES WERE REJECTED BECAUSE THEY ENTERED BROADSIDE
    93 DATA HISTORIES WERE REJECTED BECAUSE MORE THAN ONE PARTICLE WAS IN FIELD AT THE SAME TIME
  1172 DATA    RECORDS WERE REJECTED BECAUSE THEY WERE OBTAINED WITH THE CLOUD   PROBE
```

APPENDIX 17 - PROGRAM KNOLL2D MODIFICATIONS:

LOCATION	STATEMENT
Block Data (line 15)	DATA WDMM/0.06,0.24/
Subroutine VCOCAL (line 63)	VEL1 = TAS = 269.215 + 0.0832 * (ISEC - 69300)
Subroutine VCOCAL (line 64)	IF (ISEC.LT.69300) VEL1 = TAS = 260.0
Subroutine VCOCAL (line 70)	VEL1 = VEL1 * .5144
Function PSVOL (line 32)	IF (IPROBE.EQ.1) DOF=AMIN1(61.0,(3.5*ICHAN)**2)
Function PSVOL (line 33)	IF (IPROBE.EQ.2) DOF=AMIN1(261.0,(14.5*ICHAN)**2)

APPENDIX 18 - PARTICLE TYPE (PASS) CARDS

TIME INTERVAL		PMS-2D PROBE	
FROM	TO	CLOUD	PRECIPITATION
19:15:00	19:18:59	RAIN (01)	RAIN (01)
19:19:00	19:22:59	RAIN (01)	RAIN (01)
19:23:00	19:25:00	WET SNOW (03)	WET SNOW (03)
19:25:01	19:31:30	SMALL SNOW (07)	SMALL SNOW (07)
19:31:31	19:34:30	PLATES (15)	PLATES (15)
19:34:31	19:37:30	AGGREGATE PLATES + DENDRITES (17)	NEEDLES (13)
19:37:31	19:38:44	NEEDLES (13)	NEEDLES (13)
19:38:45	19:42:00	NEEDLES (13)	NEEDLES (13)

APPENDIX 19 - MODIFICATIONS TO PROGRAM PMS2D

LINE NUMBER	MAIN PROGRAM
Insert after 113	COMMON/SIXTEEN/PROB1(5),VERA41,LR1
Add to 122	,VERA41
Insert after 130	NAMELIST/TYPDAT/PROB1,VERA41,LR1
Insert after 136	READ(UNIT=8,FMT=TYPDAT) WRITE(UNIT=6,FMT=TYPDAT)

SUBROUTINE CLDA41

Insert after 67	IF (LR1.GE.4) GO TO 901
Add to 81	Statement Label "901"
Add to 170	,22,22)

SUBROUTINE LONG

Insert after 18	COMMON/SIXTEEN/PROB1(5),VERA41,LR1
Add to 28	,VERA41
Insert after 104	IF (LR1.GE.4) GO TO 901
Add to 113	Statement Label "901"
Insert after 157	IF (LR1.GE.4) GO TO 903
Add to 162	Statement Label "903"

SUBROUTINE SHORT

Insert after 15	COMMON/SIXTEEN/PROB1(5),VERA41,LR1
Add to 22	,VERA41
Insert after 34	IF (LR1.EQ.5) GO TO 901
Add to 54	Statement Label "901"
Insert after 89	IF (LR1.EQ.5) GO TO 902
Add to 95	Statement Label "902"
Insert after 102	IF (LR1.EQ.5) GO TO 903
Add to 106	Statement Label "903"

SUBROUTINE SHORT

Insert after 6	COMMON/SIXTEEN/PROB1(5),VERA41,LR1
Add to 7	,VERA41
10	Delete "+ Y(70),PROB1(5)"
Add to 9	,Y(70)
17	Delete "PROB1(5*0)"
27	Delete "LR1=2"
92	Change "DUM" to "1:05" and "10" to "VERA41"

END

2-81

DTIC